

;login:

July/August 1992

Vol. 17, No. 4

From the Editor's Desk

Maybe it's the upcoming elections. Maybe it's summer. Maybe it's negative ions in the air. Whatever it is, people are certainly talking.

For many years I have included a solicitation or exhortation at the end of many postings, articles, and editorials – "Write an article," "Debate this point," "Send in your comments and requests." The response would be maybe 5-10 replies or comments a year. Recently, however, there's been a tremendous increase in traffic on the net in both newsgroups and private mail. It's an electronic Renaissance! I hope that this constructive trend continues, with postings that are reasoned, logical, and (best of all) creative.

In this issue, I am delighted that so many members have submitted their work. Of special note is the first-ever report on a USENIX workshop, see page 4.

You'll also see the inaugural SAGE section. SAGE, the System Administrators Guild, was launched along with other potential Special Technical Groups (STGs), by the Association at the San Antonio conference.

As 'Conference Convener' for the Winter USENIX conference in San Diego, I'm hoping to identify note-takers and volunteers who can summarize sessions and BOFs. I'm looking for writers who who can distill high points so members who aren't able to attend the conference can scan the most important or provocative points from the keynote, tech sessions, invited talks, and BOFs. If you're interested in volunteering, please send me your name, e-mail address, and interests.

Of course, we're always looking for more material for this newsletter. Please contact me at kolstad@bsd.i.com with articles ranging from 200 word fillers to full blown technical discussions.

You'll note that the 'look' of this newsletter is evolving. USENIX's resident formatter, Carolyn Carr, gets the credit for these improvements. Watch over the next few months for greater readability and layout. Thanks, Carolyn!

RK

Association News

Special Technical Groups and SAGE.....	3
Micro-Kernel Workshop Report.....	4
Summer '92 Conference Report	9
Know Your Board and Staff	14
Board Meeting Summary	15

SAGE News

Mission Statement and Charter	17
Governing Board and Elections	18
Report on First SAGE Meeting.....	19
<i>Paul Evans</i>	
Working Groups.....	20
Book Reviews	22
<i>Steve Simmons</i>	
SAGE Membership Form.....	23

Features

Simmons' Laws of System Administration.....	24
<i>Steve Simmons</i>	
Name Frequency on Usenet	26
<i>Elizabeth Zwicky</i>	
An Update of UNIX-Related Standards Activities.....	30
<i>Stephen Walli</i>	
ISO Monitor Report	45
<i>Stephen Walli</i>	

Book Reviews

The Bookworm	48
<i>Peter Salus</i>	
Zen and the Art of Internet.....	50
<i>Billy Barron</i>	

Announcements

Third UNIX Security Symposium	51
System Administration Conference (LISA VI)	53
UNIX Applications Development Symposium	56
Mach Symposium	58
EurOpen Spring '93 Conference	60
Publications Order Forms.....	62
Local User Groups	65
Calendar of Events.....	67

The closing date for submissions to the next issue of ;login: is August 24, 1992.



The UNIX and Advanced Computing Systems
Professional & Technical Association

General Information

login: is the official newsletter of the USENIX Association.

login: (ISSN 1044-6397) Volume 17, Number 4 (July / August 1992) is published bimonthly by the USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710. \$24 of each member's annual dues is for an annual subscription to *login*. Subscriptions for non-members are \$50 per year. Second - class postage paid at Berkeley, CA and additional offices. POSTMASTER: Send address changes to *login*, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

Contributions Solicited

You are encouraged to contribute articles, book reviews, and announcements to *login*. Send them via email to login@usenix.org or through the postal system to the Association office. Send SAGE material to bigmac@erg.sri.com. The Association reserves the right to edit submitted material. Any reproduction of this newsletter in its entirety or in part requires the permission of the Association and the author(s).

Editorial Staff

Rob Kolstad, Editor <kolstad@usenix.org>
Ellie Young, Staff Editor <ellie@usenix.org>
Carolyn S. Carr, Managing Editor and Typesetter <carolyn@usenix.org>
Stephe Walli, Standards Report Editor <stephe@mks.com>
Bryan MacDonald, SAGE Editor <bigmac@erg.sri.com>
Michelle Dominijanni, Copy Editor

Membership and Publications

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
Telephone: 510/528-8649
FAX: 510/548-5738
Email: <office@usenix.org>

UUNET Subscriptions

UUNET Technologies, Inc.
3110 Fairview Park Drive, Suite 570
Falls Church, VA 22042
703/ 204-8000
FAX 703/204-8001
<uunet-request@uunet.uu.net>

Conferences, Workshops & Symposia

Judith F. DesHarnais, Conference Coordinator
USENIX Conference Office
22672 Lambert Street, Suite 613
El Toro, CA 92630
Telephone: 714/588-8649
FAX: 714/588-9706
<conference@usenix.org>

Tutorials

Daniel V. Klein, Tutorial Coordinator
Telephone: 412/421-2332
<dvk@usenix.org>

Supporting Members

Digital Equipment Corporation
Frame Technology, Inc.
Matsushita Graphic Communication Systems Inc
mt Xinu
Open Software Foundation
Quality Micro Systems
Rational Corporation
Sun Microsystems, Inc.
Sybase, Inc.
UNIX System Laboratories, Inc.
UUNET Technologies, Inc.

USENIX Association Board of Directors

If a member wishes to communicate directly with the USENIX Board of Directors, he or she may do so by writing to board@usenix.org.

President:

Stephen C. Johnson <scj@usenix.org>

Vice President:

Michael D. O'Dell <mo@usenix.org>

Secretary:

Evi Nemeth <evi@usenix.org>

Treasurer:

Rick Adams <rick@usenix.org>

Directors:

Eric Allman <eric@usenix.org>
Tom Christiansen <tchrist@usenix.org>
Lori Grob <grob@usenix.org>
Barry Shein <bzs@usenix.org>

USENIX Association

Executive Director

Ellie Young <ellie@usenix.org>

Copyright © 1992 USENIX Association. The Association acknowledges all trade references made herein.

login: was produced on a SUN 3/50 with Framemaker 3.0 software. It is printed on recycled paper.



Special Technical Groups and SAGE

USENIX Launches Special Technical Groups & Local Technical Groups

"USENIX is excited to have this new venue for our members to participate in ever-more focussed activities under the Association's umbrella," said outgoing President Marshall Kirk McKusick as he announced a new policy for creation of Special Technical Groups (STGs) at the San Antonio Technical conference.

STGs will provide a mechanism for all interested members of the Association to pursue technical and professional interests within a narrow focus while remaining within the larger framework of the USENIX Association.

The first STG to be chartered is SAGE (see below). This group will be concerned with systems administration issues.

At the same time the Association created a framework for the chartering of Local Technical Groups (LTGs). LTGs will provide a mechanism for Association members to conduct activities on a local level and be formally recognized by USENIX. This will encourage frequent, collegial information exchange.

Further information on the process of establishing STGs and LTGs will appear in the forthcoming issue of this newsletter.

USENIX Launches SAGE: Systems Administrators' Guild

On June 9, USENIX launched SAGE, the Systems Administrators' Guild, as the first USENIX Special Technical Group. SAGE is devoted to the advancement of systems administration as a profession. USENIX and SAGE will work jointly to publish

technical information and sponsor conferences, workshops, tutorials, and local groups in the systems administration field.

An interim board was appointed, including Elizabeth Zwicky (from SRI International) President, John F. Detke (from Octel) Treasurer, Tina Darmohray (from Lawrence Livermore National Laboratories) Secretary, and Bryan McDonald (from SRI International) Publications Coordinator. Elections will be held after the LISA Conference in October to elect a new board, which will take office in January, 1993.

Stephen C. Johnson, incoming USENIX President, said, "Our aim in launching SAGE is to encourage the recognition of systems administration as an increasingly important technical and professional specialty. Because Unix systems typically have extensive network support, Unix systems administrators are being called upon to develop and administer corporate networks that include PCs, Macintoshes, and worldwide networking. The critical nature of these tasks, and their technical complexity, makes it appropriate for USENIX to encourage the development, interchange, and publication of tools and techniques for systems administration." Elizabeth Zwicky, interim SAGE President, said, "USENIX has historically supported the systems administration community through the annual LISA Conference. We are looking forward to working closely with USENIX to expand that support, providing more avenues for the professional and technical growth of the field."

If you would like to join SAGE, please use the enclosed application form included on page 23 of this issue.

Micro-Kernel Workshop Report

Report on the Workshop on Micro-Kernels and Other Kernel Architectures

Seattle, WA, April 27-28, 1992

by Peter S. Langston
<psl@bellcore.com>

The first USENIX Workshop on Micro-Kernels and Other Kernel Architectures was held on April 27-28 in sylvan Seattle, Washington (a.k.a. the "Emerald City" for the same reason that it is a.k.a. the "Rain City"). With 334 attendees, more than three times the anticipated number, this was not so much a workshop as an SRO conference – literally every seat in the huge meeting room was taken and some people even had to be turned away. The rest of the title of this workshop/conference also managed to cause controversy. While some thought that the title should logically have been reduced to "Workshop on Kernel Architectures," others thought that the workshop was probably aimed at comparing and contrasting existing micro-kernels and their macro-kernel counterparts and therefore should have been called "Workshop on Micro-Kernels vs. Other Kernel Architectures." Still others questioned the use of the term "micro-kernel" to describe systems that require several megabytes of memory to operate a light switch ("Does this mean that MVS was really an early micro-kernel?"). In any case, the persistent inclusion of phrases such as "and other kernels" betrayed the catholic intentions of the organizers.

Monday, April 27

As advertised, the first day's sessions dealt with introductory talks on currently important micro-kernels "and other kernels." After the opening remarks from Program Chair Lori Grob, five one-hour overview talks were presented.

Robbert van Renesse (Vrije Universiteit/Cornell University) presented a "Short Overview of Amoeba" which was an update on a talk that has been given a few times before. His talk featured some interesting new slides including one that was the basis for Figure 1 shown here (with the addition of an NT column and minor editing by me).

Rich Draves (Carnegie Mellon University) gave a talk on "Microkernel Operating System Architec-

ture and Mach" that was characterized by one of the other paper presenters as "the most realistic Mach talk ever." That evaluation may have been influenced by a section of the talk dealing with difficult decisions and things that they might do differently next time.

Dave Presotto (AT&T Bell Laboratories) described "Plan 9, A Distributed System." Aside from describing the ideas and implementation of Plan 9 (simplification and minimalism expressed as taking a few good ideas and using them to extremes), Dave also described a luxurious backup system and provided some aphorisms: "file systems are cool," "name spaces are cool," and so on.

Marc Rozier (Chorus Systèmes) gave an "Overview of the Chorus Distributed Operating System" that placed it both historically and technically. Much of Chorus' terminology predates the current crop of buzzwords, making the associated paper refreshingly free of them. But not to be left behind, Marc's talk joined the rush into the mid-90's with a "nano-kernel" (== _ Real-time Executive). Marc also described "COOL" (Chorus Object Oriented Layer) but missed the chance to aphorize that "Chorus's object oriented layers are cool."

David Cutler (Microsoft Corporation) spoke on "Microsoft Windows NT" giving a broad overview ranging from Microsoft's systems strategy and market perspective through architectural issues to time and space measurements. David made the observation that NT is hardly a micro-kernel and must be the "other kernel architecture" mentioned in the workshop title. The next session, chaired by Dag Johansen (University of Tromsø), was on "New Architectures" and consisted of two papers.

Jonathan Walpole (Oregon Graduate Institute of Science and Technology) described "Modularity and Interfaces in Micro-Kernel Design and Implementation: A Case Study of Chorus on the HP-PA Risc." This port of Chorus to the HP PA-RISC workstation took a year to do, uncovered common but inefficient operating system interface architectural assumptions, and illustrated the tradeoff between micro-kernel modularity and performance. Even better, the first sentence of the paper's abstract actually answers one of the ques-

tions raised by the workshop's statement of purpose.

Toshio Okamoto presented a paper entitled "A Micro Kernel Architecture for Next Generation Processors" outlining a design for an OS kernel that takes advantage of the large address space of new processors. Three design features are postulated: single virtual storage (no context switch address remapping), one-level storage (files, libraries, etc. are all parts of the single address space), and fine-grain memory protection (PTEs and two kinds of ACLs implemented by a fancy

For the panel members I have used initials (so only really smart people will know who said what).

PH: It's all lies! It's all the same bloat as a Unix kernel - so why is it "micro"?

RD: You can throw away the Unix part of the bloat...

PH: But then it's not useful!

JL: If you want to make money it will have to

	Amoeba	Mach	Plan 9	Chorus	NT
Architecture	Centralized processor pool	Symmetric	Centralized processor pool	Symmetric	Symmetric
Model	Object-based	? (whatever)	File-based	Object-based	Object-based
Communication	RPC multicast	Message + RPC	Streams + file system RPCs	Message + RPC + unreliable multicast	LPC + RPC
Naming	Capabilities + directory service	Port rights + naming service	File name space (directories)	Capabilities	Unified name space per machine
Protection	Capabilities	Port rights communication based	Owner/group/other (owner can be a set of users)	Capabilities	All objects protected with ACLs
Light-weight processes	Yes, kernel-scheduled	Yes	No	Yes	Yes, kernel-scheduled
Unix support	Slow source emulation	Yes	Almost exactly Unix with library level Posix	Yes	Posix support
Distributed applications (across net)	Excellent support	OK	Not really	OK	Yes, RPC based
Multiprocessor support	Yes	Excellent	Great UMA (SMP) support	Yes	Excellent
Virtual memory	Segments	Paging	Paging	Paging	Paging
Fault tolerance	Replicated services	No explicit support	Great backups	Dynamic reconfiguration	Mirroring, striping, duplexing, & others

Figure 1

MMU). Any paper that shows the subroutine call as a new way to do message-passing gets my vote.

The final session of the day was the Micro-Kernels Panel Session moderated [sic] by Peter Honeyman (University of Michigan). The panelists were: Dave Presotto, David Cutler, Rich Draves, Jim Lipkis (Chorus Systèmes), and Robbert van Renesse. It is not unusual for the moderator of such a panel discussion to have to calm down the panelists and act as peacemaker; that did not happen. What did happen? Well, here's what I was able to write down. I started out indicating all the places where there was general laughter or applause, but there were so many that I had to give it up. "Floor" is used to indicate a question or comment from "the floor."

have Unix.

DC: or DOS. ... [discussion of some Plan 9 port that was done in 7 days - mostly the time it took Ken Thompson to port the C compiler.]

DC: Have you guys got any more of those porting guys? I'd like a couple.

PH.: I don't think you can afford them, Dave. [D.P. talks a lot and teases Chorus for their diagrams] ...

PH: Dave, what does NT stand for, anyway? It certainly couldn't be "New Technology" ...

DC: If you've ever seen the inside of DOS you'd

see why NT is New Technology.

PH: I'd like to be the first to welcome Microsoft into the 1970s. ...

PH: Everyone but Plan 9 claims to be a virtual "porting machine."

... [PH asks a confusing question of JL and then tries again and makes it coherent and gets a careful, coherent, and unamusing answer. This is followed by an unanswered, but much more amusing question.]

...[DP berates RD for Mach trying to be all things for all people and wanting to make a platform that then requires everything else (e.g., Unix emulation) to be added on.]

JL: The world outside this room doesn't care about minimality and cleanliness.

DP: Now you're getting to the important point—none of this really matters! ... [An audience member asks a question about whether any of the so-called micro-kernels can run in 8K of memory]

RR: Amoeba will run on an 8k machine - with the right ifdefs. [JL claims that minimality was investigated as a goal (for Chorus) and found "not to be a win" so all the things that were removed were put back in.]

JL: Minimality itself is not much of a goal.

Floor: [The questioner starts with a long list of the problems that beset developers]... So what have you guys done to help me with these problems? [A loooong silence...]

PH: Well, I thought it was a bullshit question, too. ... [A little later another audience member asks ...]

AM: Er, my question is kinda like the preceding one...

PH: Well forget it! I don't want to talk about it anymore!

DP: Perhaps, if you could rephrase it as a personal insult? ... [Mike O'Dell asks a question from the floor about distribution involving many pieces of bread, peanut butter, and a squeegee that no one understands but JL (maybe). So JL suggests a related question (that he *can* answer) and answers it. Mike seems mollified.] .

Floor: What do these systems do when faced with data rates of a terabit a second?

DP: Does "choke and die" mean anything? [general agreement] We're still limited by our inter-

faces to about 10 megabytes per second. ...

Floor: I have two questions. Blah, blah, ...[I didn't write down the first question]... blah?

PH: That's a bullshit question; what's your other one?

Floor: [Unfazed] Okay; Plan 9. How do you know what something's called if everything can have its own name space?

DP: ...by convention... [he gets onto the subject of catching all filesystem references]... The ability to do that, the ability to circumscribe the world, ... er, ... to circumscribe the world is immensely powerful...

...[there follows a fairly long discussion over whether Plan 9's lack of structure is a Good Thing. DP's apparent willingness to admit the possibility of being wrong creating something of a feeding frenzy among the other panelists.]

PH: Well, that's it. Goodbye.

The reception that followed was distinguished only by an unusual surfeit of blue sky outside the picture windows and an unusual deficit of beer from the excellent local micro-breweries. The appearance of six bottles of Red Hook early in the night (hotel leftovers) only served to whet appetites that could not be satisfied in the hotel.

Tuesday, April 28

The first session on the second day was called "New Systems" and was chaired by Robbert van Renesse. Four papers were presented.

Charles Landau (MACS Lab, Inc.) talked about "The KeyKOS 9 Nanokernel Architecture." Development of this nanokernel system began in 1975 and the system was in production use by 1983. It can run in 100 kilobytes of memory and a subset of MVS has been ported to the KeyKOS platform. Designed to favor reliability and security over performance, the system requires extraordinary measures to set capabilities at initial startup, but once set they are "persistent" and can be retracted only by prearrangement. This makes a development problem when a test system gets "weird;" even pulling the plug doesn't fix it because it is "persistently weird."

Dan Hildebrand (Quantum Software Systems) gave "An Architectural Overview of QNX." This new system has only existed since 1982 and was (according to Intel Corp.) the first multiprocessing OS on the PC. The latest version is POSIX compliant and only requires 6.8K bytes of memory for

the micro-kernel, but would require nearer 100K for a minimal, a.k.a. "light switch," OS, (big enough to be a nano-kernel, I guess). Audience questions concerned clock synchronization on the LAN and plans to port to a RISC machine (no).

E. Douglas Jensen (Digital Equipment Corporation) spoke on "An Architectural Overview Of The Alpha Real-Time Distributed Kernel." This amusing talk about the distributed thread, real-time, OS kernel joint project involving Concurrent Computer Corp., DEC, and the Open Software Foundation contained numerous pithy quotes – "Real Fast" is not "Real Time" – "The security guys are seriously anal retentive" – "There's nothing micro about Alpha." Strangely enough, in support of the last statement he mentioned that the source code was 20,000 lines of C, the same number claimed for the KeyKOS nanokernel.

W.E. Kuhnhauser (German National Research Center for Computer Science) talked about "Performance of the BirliX Operating System." While the paper characterizes BirliX as "an operating system for distributed, secure, and fault-tolerant applications" the speaker pointed out that it may be viewed as a "persistent object management system" and not a micro-kernel in any case. On the other hand, this is probably the newest of the new systems that were presented in this session.

During the break before the next session, the team that had spent the previous evening investigating Micro-breweries and Other Brewery Architectures made an appearance to give a report (the principal investigator's coloration could only be explained as a tribute to the Emerald City). Following the break, a paper session entitled "Lessons Learned," chaired by Edward Lazowska (University of Washington), presented three papers.

Jun Nakajima (Fujitsu Laboratories Ltd.) described "Multimedia / Realtime Extensions for Mach 3.0" making some interesting comparisons between Mach 2.5 and Mach 3.0 in the process. He divided multimedia devices into two types – response-time-sensitive (event-driven) and response-time-insensitive (deadline-driven) and showed how extensions to include "realtime threads" and a "temporal paging system" handle them.

Henry Massalin (Columbia University) spoke about "Reimplementing the Synthesis Kernel on the NeWS Workstation." Synthesis breaks most

of the rules: it is written entirely in macro assembler, the kernel includes self-modifying code, it is blindingly fast (as are the programs that run on it), it is small (a minimal kernel runs in 16K RAM and 16K ROM), and is not called a micro-, nano-, pico-, or femto-kernel. Henry played a recording of some music produced by software synthesis. He mentioned that a keyboard note generator program takes 720 micro-seconds to (1) sense a key press, (2) create a thread, (3) attach the thread to the audio output, (4) start executing the thread, and (5) produce the beginning of the sound output. Henry also described some clever solutions to cache concurrency problems encountered by machines executing self-modifying code. Quincy and his daughter Emily appeared briefly and said "qua" encouragingly to the audience.

William Davenport (Digital Equipment Corporation) presented "A Model and Prototype of VMS Using the Mach 3.0 Kernel." Modeling VMS took 9 months; prototyping the VMS model took another 3 months. A plea for a native mode Mach debugger was made (with agreement from the audience). After implementing 46 of the 250 VMS system services several VNS utilities were found to be runnable. Conclusions were drawn: micro-kernel technology is cool and multi-server technology is cool, but performance is probably a casualty.

Lunch was uneventful except that we got to see Historic Pike Place Market, ate a lot of Mexican food, and drank fluorescent Mexican sodas (to the horror of the aforementioned micro-brewery test team captain).

Program chair Lori Grob was also session chair for the following session "Experience and Observations I" comprising three papers.

Brian Bershad (Carnegie Mellon University) decried "The Increasing Irrelevance of IPC Performance for Microkernel-Based Operating Systems" while new Seattle resident Rick Rashid turned the slides. Four points were advanced: IPC has gotten faster than other stuff; caches, not address spaces, determine performance; all data does [sic] not need to go through the kernel; all services do not need a hardware firewall. The question period was initiated with the reminder that an unwritten rule disallows the slide turner from asking questions.

Jochen Liedtke (German National Research Center) presented a paper on "Fast Thread Management and Communication Without Continuations" that describes the operating system L3,

argues for the relevance of IPC and concludes that (1) IPC can be implemented really fast; (2) continuations will not support this job; and (3) availability of fast IPC changes programming behavior. Confused questions ensued.

Jim Hamrick (Unisys Corporation) discussed "Experience with SVR4 Over Chorus" and stressed that the project was one of very few involving commercial product development with micro-kernels rather than academic research on them. Twenty-two people spent eighteen months bringing the project to completion. The initial requirements are met and the system is stable.

The final break of the conference passed with no noteworthy occurrences. The chair for the last session "Experience and Observations II" was Jim Lipkis.

Randy Dean (Carnegie Mellon University) pointed out that his talk would be different from the paper "Data Movement in Kernelized Systems" in that the paper strives to describe Chorus and Mach side by side while the talk just focuses on their similarities, which include: VM central caching, an external mapper, fast and reliable IPC, and a trap redirection mechanism. He concludes that kernelized systems are here [but are they cool?] and good file system performance is possible.

Marc Shapiro (INRIA) gave a design report entitled "Distributed Abstractions, Lightweight References" in which a library of useful abstractions structured as fragmented objects and protocols to support lightweight, robust, uniform, garbage

collected, distributed references are proposed as amendments to current operating systems designs in place of the more "heavyweight" ports, pipes, and sockets.

Robbert van Renesse presented "Reliable Multicast between Micro-Kernels," describing a re-implementation of the ISIS system designed specifically to take advantage of micro-kernel technology and fill in some gaps in current micro-kernel support (e.g., cross-network communication and failure detection). One of the goals is to make the ISIS system "FTTPable" and examples dealt with the netnews-like "ISIS news groups."

Michael Stumm (University of Toronto) gave the final paper, "Designing a Scalable Operating System for Shared Memory Multiprocessors." This paper proposes a structuring technique based on clustering to solve problems of scalability in multiprocessor operating system design.

As the second day came to a close and questionnaires were handed out, attendees had a chance to look back over the two days and evaluate the workshop design. The initial overview sessions established a basis of reference for the later discussion (and disabused the attendees of any notion that the term "micro-kernel" implies something about size). The following papers were both interesting and well-presented and the panel session was ... well, interestingly presented. This timely workshop dealt with a topic that, as the attendance attests, is a real "hot button." The organizers (and the program chair, Lori Grob, in particular) deserve kudos for a job well done.

Summer '92 Conference Report

Best Paper Winners

Doug Moen of Sietec Open Systems Division was given the Best Paper Award for *A Discipline of Error Handling*.

Mary Baker and Mark Sullivan, UC Berkeley authors of *The Recover Box*, won the Best Student Paper Award.

Both papers have been published in the USENIX Summer 1992 Technical Conference Proceedings.

Works-in-Progress Report

Organized by Lisa A. Bloch, Sun User Group and coordinated by Peg Schafer, BBN Systems

A Hybrid Performance Model for NFS File Servers

David N. Williams, Ericsson Network Systems, Richardson, TX <exudnw@exu.ericsson.se>

In this session we will report on a Hybrid simulation model of NFS client-server transactions.

Our current environment consists of over 400 diskless SPARC workstations supported by nine Sun 4/490 active servers. Benchmarks combined with trial and error were the prime methods used in arriving at the current configuration.

A number of benchmarks exist to assist in selecting and tuning NFS servers, but benchmarking has its perils and limitations. Vendor-supplied benchmark numbers are frequently suspect, and not every organization has the resources or skills required to achieve accurate and meaningful results. Even after spending extensive time benchmarking a server, the results may not provide sufficient information on how it will work in a specific environment.

A discrete event simulation model of the NFS client-server relationship has been built which provides an approximate model of existing or proposed client-server configurations. The model allows for flexibility in changing parameters and does not require the investment in time, and possibly money, that comes with benchmarking.

Phonestation: Moving the telephone onto the Virtual Desktop

Stephen Uhler, Bellcore
<sau@bellcore.com>

The PhoneStation system is a tool that allows a Sun SPARCstation to control an ordinary telephone line. It consists of: 1) a micro-controller that interfaces a telephone line with a SPARCstation via a serial port and the audio connector, 2) a software library for the SPARCstation that provides telephone interface control, digital signal processing (e.g., touch-tone detection), and text to speech conversion, and 3) a TCL based language for writing telephone applications.

As an example, the system can be used to integrate answering machine functionality into the workstation environment. Voice messages appear as ordinary electronic mail and are played through the SPARCstation speaker. If mail is read from a dumb terminal, the PhoneStation system places a call to a user specified telephone number and plays the voice portions of any messages.

Texas: An efficient, highly compatible persistent object store using pointer swizzling at page fault time

Vivek Singhal, University of Texas at Austin <singhal@cs.utexas.edu>

Texas is a persistent object store that implements huge address spaces efficiently on standard hardware. Pointer swizzling (address translation) at page fault time converts the pointers in a page from a long format into normal, hardware-supported virtual addresses when pages are brought into memory. This translation is transparent to compiled programs, allowing the use of existing compilers with little or no modification. Modern UNIXes such as SVR4 and OSF/1 provide the necessary control over virtual memory with no modifications to the operating system.

Gumby: The portable, high-performance file system that rides on the back of your Pokey file system

Sheetal V. Kakkad, University of Texas at Austin

Gumby is a simple log-structured file system built on a normal UNIX file system. The file system is built inside a single UNIX file, requiring no

dependencies on underlying disk geometry, so it is quite portable. Log structure avoids the use of a single "home" disk block for a logical file block, allowing any block to be written anywhere. This optimizes the file system for use with large RAM caches, which tend to absorb most reads and increase the proportion of writes. We intend to experiment with reordering read-only pages as well, to dynamically increase locality and reduce seeks caused by read misses.

Knowledge-Based Systems Construction in C++

Vladimir Bacvanski, Aachen University of Technology, Germany
<vladimir@rwthi3.informatik.rwth-aachen.de>

The examination of an applicability of appealing techniques from object-oriented software engineering to knowledge-based systems domain is discussed, focusing on the promising role of C++ in this context. The entrance of expert systems into real industrial application arena has uncovered weak points of the current knowledge-based systems technology, especially the incomprehensibility, poor performance, and inability to integrate with non-knowledge-based systems.

The use of C++ for building technical expert systems should provide one possible framework for overcoming the current deficiencies. The code of a knowledge representation language is translated into C++, bringing the possibility to use knowledge-based techniques while remaining in the well known environment, so that developers do not have to abandon all their skills and move to expensive and incompatible specialized artificial intelligence workstations. Moreover, the combination of multiple paradigms (object-oriented, procedural, and the rule-based one) in the C++ framework produces a synergetic result.

A new multi-paradigm system architecture is examined together with mechanisms which diminish the impedance mismatch between object-oriented knowledge and non-knowledge-based systems, providing interchangeability of objects which follow different paradigms. The object-oriented paradigm is used not only to model the applications, but the system's internal components as well. The correspondence between different constructs from the object-oriented and knowledge-based systems will be investigated, showing that it is possible and profitable to model knowledge-based systems with a set of C++ classes.

Development of an event based debugger with source level capabilities

J. G. Posthuma, J. Scholten, J. G. Wijnstra;
U. Twente, Netherlands
<posthuma@cs.utwente.nl>

Finding a bug in an application is time consuming and expensive. For parallel applications, debugging is even harder. The behavior of parallel applications can only be understood by looking at them with great abstraction. Only specified events of the system should be presented to the user. But such events only give a hint where a bug could be. After this hint, the user has to look with greater detail. He should be able to specify both events of higher abstraction (for example communication) and source level debug events.

Events are the basis of the debugger. An event is generated each time an important point is reached in the execution. These points can be specified by the programmer. An event will often be used to indicate a place where interaction between two processes takes place, since interaction is an important aspect of parallel applications. The events of all processes are merged into one event stream. This stream can be used directly or stored in a database for later use. For a long running or multi-process application, the event stream can be quite voluminous. That is why a number of tools are provided to make the event stream data more manageable. The programmer has the possibility of reducing the complexity by specifying filters, which remove those events in which the user is not interested.

Another important tool is the behavior recognizer. A recognizer matches behavior as specified by the user in terms of events against the event stream generated by the application. It is not only possible to specify the expected behavior, but also the behavior that is not allowed. Recognizers can be used in a number of ways. First of all they can be used to trace down the places where the specified behavior is violated. Secondly, recognizers are useful to summarize the behavior by replacing a number of events from the stream with one new high level event. This allows the programmer to analyze the system at different levels of abstraction. A third possibility is to use recognizers to specify interesting points in execution where some action must be performed, like a request for process status. This last option is only possible for run time debugging. The recognizers are also useful for analyzing events in the database.

The debugger is a mixture of an event based debugger, source level debugger, and a behavior recognition system. The event based part of the debugger concentrates on distributed aspects of the application and can be used during or after execution. It is not always possible to debug an application when using the event based approach. That is why the normal source level debugger will be integrated in the design. In the future, other features may be added, in order to make it a complete debugger for distributed applications.

MACH on a Physically Secure Crypto Coprocessor

Elaine Palmer, IBM Research Division, Yorktown Heights, New York
<epalmer@watson.ibm.com>

I will present an overview of a secure cryptographic workstation coprocessor. The prototype hardware, named 'Citadel', was completed in 1991 at IBM's T.J. Watson Research Center. It can perform DES encryption and decryption of data at high speeds.

We envision its use in three possible application environments:

1. It can be embedded in a communications controller (wire or fiber) to encrypt or decrypt some or all network traffic.
2. It can be embedded in a disk controller to encrypt or decrypt selected files or an entire disk.
3. It can be installed as a secure, general purpose crypto coprocessor on a microchannel bus. In this environment it can be used to encrypt or decrypt passwords, authenticate users and requests for resources, encrypt or decrypt transactions, or process sensitive data from the main host processor.

In all three applications, encryption and decryption does not degrade the throughput of its host device. The host for our prototype is an IBM PS/2.

The hardware and kernel software are designed to operate in a physically secure package. If the package detects tampering, it responds by erasing its encryption keys and other secure memory. The system software is a small, multitasking microkernel, Mach 3.0, from Carnegie Mellon University. I will discuss the advantages and the disadvantages we've encountered in using Mach for this project.

BOF Report

by Rich Salz

Open Software Foundation

<rsalz@osf.org>

A good BOF can be one of the best things to do at a USENIX Conference – it can keep you going all night.

BOF stands for Birds of a Feather, as in "flock together." It's an informal gathering of people who are interested in a particular area. Many BOFs are scheduled before the conference and announced in the conference schedule. Others are "scheduled" on site, and announced by posting a notice on the general message board. BOFs are not part of the standard conference track. They are generally held after-hours and anyone can attend. Many of them "adjourn to the bar," where the discussions can go on for hours.

This Summer Conference's BOF topics included: a discussion of Standards (POSIX et al), Distributed Systems Administration, Gays in computing, EFF, FSF, Usenet, NNTP, UUNET, Ultrix, Alpha, BSD4.4, BSDI, and Obfuscated C. I think one of the best things to do is to go to an area that's new to you. It's a great way to get practical knowledge in an informal setting, and a good way to meet experts in the field.

Unfortunately, I didn't do that this year: I went to the Usenet-related BOFs because I "had to," and others because I wanted to get a status report. So, while there were no doubt lots of good things happening at FSF, EFF, Distributed Systems, and so on, I can't tell you about them.

Tuesday: News Software & USENET

This BOF was run by Henry Spencer and Geoff Collyer. It started with an update on C News. The Performance Release (including much work done by Geoff for UUNET) is out, and the Clean-Up Release won't be out for "quite a while." This next release will have a revised source tree, and (to the cheers of the crowd) most of the build work will be done directly in the Makefiles.

News volume is still doubling yearly, and the growth in newsgroups is (apparently) causing problems for some newsreaders, most noticeably when sorting the *active* file or deleting a newsgroup. Newsreader writers, beware: the net is growing faster than you think!

The volume and newsreaders then led into a discussion of threads programs like trn. Geoff is

thinking about looking into the issue of threads databases, saying "mthreads must go." On the other hand, we did get to hear the only nice thing Geoff has ever said about NFS: "for reading news, NFS is pretty good." There was also talk about changing the news filesystem format, to which Geoff replied "*fix namei*."

Bruce Jones, from the School of Communications at UCSD, is doing his doctorate on the growth of Usenet. He has Henry's old tapes from the start of Usenet and is trying to gauge the interest in getting a CD-ROM of, say, the first A News "Car for Sale" ad. If you're interested, send email to <bjjones@ucsd.edu>.

Stan Barber spoke about the next release of NNTP. The client and server code has been split into pieces and the client code is in beta-test. It's already been ported to some PCs. The new server should go into alpha-test in July. If you have some new feature or bug fixes, let Stan know. In particular, if you can help make it work well with C News he'd like to hear from you. Stan can be reached as <nntp@tmc.edu>.

Wednesday Night BOFs

On Wednesday I attended the standard Usenet and hackers BOF track: UUNET, Obfuscated C, BSDI, and 4.4BSD. Even though each was only an hour long, this was a long night.

Unfortunately, I missed most of the UUNET BOF. I wandered in during a discussion of Alternet (UUNET's commercial IP network, no traffic restrictions). People are interested in low-cost methods like dial-up IP service. Rick Adams mentioned a bit about how the FBI is a customer. People concerned about the FBI reading netnews should make a reality check: the FBI wants to catch serial killers, they couldn't care less about obnoxious netnews postings!

UUNET has also written another version of UUCP. BSDI has licensed it, and all UUNET customers will probably be able to get it, too. The most interesting thing about the UUNET UUCP is that you can replace the front-end configuration files so that it looks like whatever version of UUCP you want it to. Only BSD is supported, but HDB is an obvious next choice.

Every year Landon Noll asks the people of Usenet to send him the most twisted C code they can write, and in the spring and summer he and his group evaluate the results and pick the best (or is it the worst?) they can find. No program could be more than 1,536 bytes of non-whitespace, and no "cc" line could be more than 256 bytes. Lots of whitespace was allowed this year,

which made most of the programs a little less fun to look at. For the first time, there were more non-US winners than those from the United States. Every year, this is one of the best BOFs: it's very technical, in a weird sort of way, and it's very funny.

I also detected a decided "tools" bent to this year's winners. It would have made a nice contrast to the FSF BOF. While GNU software does lots of nice things, nobody will ever say it's small. At the Obfuscated C BOF, however, we got to see a chess program (written by Vern Paxson, the author of *flex*) that reportedly held its own against GNU Chess. There was also a *make*-like program that had some novel features. Both of these listings could fit on a single page!

The full results will be posted to the net (in comp.lang.c, misc.misc, and other places) in a month or two. Landon also warned people that he and Larry Wall are working on an obfuscated Perl contest, which many in the crowd thought was kind of redundant.

Berkeley Systems Design, Inc., (BSDI) is a new start-up that is selling BSD operating systems for the 386-family of machines. It's a small company, still struggling to meet their weekly payroll. For about a thousand dollars, you get the full source code to BSD, X, NFS, and other tools – and binaries to run it on your IBM PC or clone. This was the most overtly commercial BOF I attended: Rob Kolstad is an entertaining speaker, but it was clearly a vendor presentation. It gave information people clearly wanted to hear, however: the room was packed. The part I found most interesting was that USL (the branch of AT&T in charge of Unix) is suing BSDI. While you can never be sure when lawyers are involved, it would seem that they are taking exception to the claim that the Berkeley "Net II" release, upon which BSDI's product is based, is unencumbered. I'm guessing that BSDI was picked because they are the first commercial venture that hasn't bought some sort of license from AT&T. For more information on BSDI, contact <info@bsd.com>.

The last BOF of the evening was the 4.4BSD BOF, led by Kirk McKusick and Keith Bostic of Computer Systems Research Group. The schedule said that this would include a report on the release schedule for 4.4BSD. This was very unusual as the CSRG folks from Berkeley have never previously announced their release schedule. Anyhow, 4.4 will be available in two formats: 4.4 and 4.4 "light." The former will require an AT&T license; the latter will contain only the freely-redistributable source code. This will be more complete than earlier free releases, but will

still need some work on the kernel. Both the alpha and the final release will be available in both formats.

4.4 will have lots of filesystem features: 64 bit file-sizes (using the *longlong* datatype), NFS "leases" that make NFS more efficient and robust, stackable filesystems (similar to what David Rosenthal discussed at Baltimore; the BSD work comes from UCLA and the Ficus project), */dev/fd*, the log-based filesystem (from Sprite), and so on. It will also make *uid* and *gid* be 32 bits, further changing the *stat* structure. These changes will all be in the alpha release because they involve changes in the system interface. The final release will have new TCP/IP work from Van Jacobson, the Berkeley streams package, and probably a new virtual memory system (from Mach). It will also contain as many documentation updates and bug fixes as possible. Sun has donated their shared library architecture, and that may also be a part of 4.4. I can't read my notes at this point, but I think the supported architectures include the Sparc, HP9000, Tahoe, and others.

The bad news is that once 4.4 is solidly out the door, the CSRG Group is shutting down. They explained that it is hard to get more funding, the University is using BSD less, it is too big for the current group to develop, and that the past year has not been fun: too much politics and name-calling. It's probably safe to say that the workstation industry would not have happened without BSD, and that many of us would be doomed to be filling out RPG II forms in dimly-lit cubicles. Thanks, guys!

Thursday BOFs

Thursday night is always a questionable night for BOFs because things are always scattershot after the USENIX reception. This didn't stop me from

scheduling the third Usenet-type BOF of the conference, however. This one concentrated on NNTP. The NNTP protocol is being revised by an Internet Engineering Task Force committee. Most of the revisions are related to supporting batching and other facilities for low-speed links. The current draft is available for FTP from turbo.bio.net in `~ftp/ietf-nntp`. The group is not concentrating on facilities for news-readers. There is an unofficial group working on that; send mail to [<David_ascher@brown.edu>](mailto:David_ascher@brown.edu) if you are interested in that area.

Stan Barber gave another preview of upcoming NNTP releases and asked for some feedback on specific changes to the client-side *inews* that is part of NNTP. This led to some discussion about news headers. There were lots of questions for me about INN, my Usenet/NNTP implementation. I mentioned it at last year's BOF and presented a paper on it this year so people were fairly curious. By the time you read this, the software should be available, however, so I won't take up any more space on self-aggrandizement.

One last-minute BOF that was held on Thursday was for archive maintainers. This group is starting a very exciting project to make a universal card-catalog for software available on the net. Many of the people involved -- Rich Morin, Ed Vilmiatti -- have lots of experience with public archives, and it sounds like they have a good plan of attack. For more information, contact cfcl!rich.

Well, that's it. I hope you thought this useful, and that it spurred your interest to become a full-fledged USENIX BOffer.

Know Your Board and Staff

Board of Directors



Stephen Johnson,
President



Mike O'Dell,
Vice President



Evi Nemeth,
Secretary



Rick Adams,
Treasurer



Eric Allman,
Director



Tom Christiansen,
Director



Lori Grob,
Director



Barry Shein,
Director

Staff



Ellie Young,
Executive Director



Diane DeMartini,
Administrative Assistant



Carolyn Carr,
Publications Manager



Judith DesHarnais,
Conference Planner



Patrick Dunne,
Administrative Assistant



Marilynn Allemann,
Conference Associate



Toni Veglia,
Administrative Assistant



Andrea Galleni,
Administrative Assistant



Rob Kolstad,
login: Editor



Daniel Klein,
Tutorial Coordinator



Cynthia Deno,
Vendor Display Coordinator

Board Meeting Summary

by Ellie Young

Executive Director

Below is a summary of the actions taken at the regular quarterly meeting of the USENIX Board of Directors held in San Antonio, Texas, on June 7, 1992.

Attendance

Rick Adams, Ed Gould, Rob Kolstad, Kirk McKusick, Sharon Murrel, Evi Nemeth, Michael D. O'Dell, Barry Shein, Ellie Young, Judy DesHarnais, Dan Klein, Eric Allman, Tom Christiansen, Peter Collinson, Diane DeMartini, Lori Grob, Steve Johnson, Greg Rose, Elizabeth Zwick

Summer '92 Conference

Adams reported that everything seemed to be on track. DesHarnais said that attendance from California was below normal, and total attendance would be 1,050. Klein noted that for this conference, the ratio between the number of tutorial seats sold to overall conference attendance is higher than in the past. It was suggested that he look into the feasibility of adding continuing education credit for tutorials.

Microkernel Workshop

Grob would submit a proposal for another event in the future, with half-day tutorials on different systems, with consideration that it be located next to SEDMS next Fall.

File Systems Workshop

Nemeth reported that the technical quality was good, the sessions were well-attended, Peter Honeyman did an excellent job, and the Ann Arbor site was good but difficult to get to. There was discussion that this event should have been called a symposium. It was decided that in the future that a limited-attendance event (e.g., requiring position papers) be called a workshop.

Chair for Winter '94 Conference

Young said she had received two proposals (one from Jeff Mogul and another from Clement Cole). It was suggested that Cole's proposal might be better as a workshop because of its timeliness, and Young was asked to discuss this possibility with him. It was decided to accept Mogul's proposal to chair this conference.

BOFs Screening

It was decided that screening BOFs for technical content was no longer necessary.

Book Program

O'Dell and Young reported that Jim Waldo had agreed to serve as editor for the USENIX papers on C++, and Simon Kenyon would serve as editor of a series on kernel technology. There were several other projects in the works, and they hoped to have at least four titles to launch this series next year, with MIT Press as the publisher.

IR Report.

Collinson announced that he would be stepping down as the USENIX Institutional Representative after the October '93 POSIX meeting, and offered to assist in the hiring of a replacement. Collinson would publish a request for proposals and field questions concerning the position in the coming months. A committee of Allman, McKusick, and Collinson would review the proposals and make a recommendation to the Board at the next meeting.

Policies

It was decided to add another section to the current attendee list that is sorted by zip code so that people can locate others who are geographically close. It was decided to postpone making a decision on having the list available online until the data from the attendee survey was available. Murrel's proposed document (replacing Sections 6.2 and 6.3 of the current policies document) which would divide the Association's accounts into two funds, thereby making them easier to manage, was accepted.

Honorary Membership

It was agreed to make Lew Law our second honorary lifetime member.

STGs/SAGE

It was decided to adopt Adams' SIG proposal document with three amendments. The first amendment clarifies that 1) USENIX has complete responsibility for the LISA conferences, 2) that all members of the SIGs must be members of USENIX, and 3) that the name SIG is changed to STG (Special Technical Group).

It was agreed to adopt Johnson's amended resolution creating SAGE as an STG, as follows:

1. That the USENIX Association launch a special technical group to be known as USENIX/SAGE (Systems Administrators' Guild). This organization will engage in education, develop standards of excellence, recognize those who attain such standards, and promote work and propagate knowledge that advances the systems administration profession.

2. That the USENIX Association further appoint an interim governing board for USENIX/SAGE, consisting of:

Shoshana Abrass,
Tina Darmohray, Secretary
Arnold deLeon
John F. Detke, Treasurer
Paul Evans
Laura deLeon
Bryan McDonald, Publisher
Paul M. Moriarty
Arch Mott
Bjorn Satdeva
Steve Simmons
Pat Wilson
Elizabeth Zwicky, President

3. That the USENIX Association immediately accept membership applications to be a member of SAGE. The pro-rated dues for the remainder of 1992 will be \$25. All SAGE members will also have to be USENIX members.

4. That the USENIX Association will administer an election for a 9-member SAGE board, said election being concluded so that the new board can take office after the 1992 LISA conference.

5. That the 1992 LISA conference base price will reflect the cost for SAGE members. USENIX members who are not SAGE members may check a box which adds \$25 to the cost and makes them SAGE members. People who are neither USENIX or SAGE members check a box which makes them members of both organizations.

6. That the Executive Director establish a membership category for SAGE membership, and maintain office services (such as mailing lists and financial accounting) for SAGE as well as USENIX. Moreover, the Executive Director will ensure that SAGE and the SAGE board members are covered by liability insurance.

7. That the USENIX board will appoint a formal liaison to the SAGE board, expecting that the

SAGE board will likewise appoint a USENIX board liaison.

8. For the 1994 LISA conference, the program chair for LISA will be picked by a four person committee, two members of which will be appointed by the SAGE board and two members by the USENIX board. This committee will also work with the USENIX Tutorial Administrator to ensure relevant, high quality tutorials at LISA.

9. That a regular section of *login:* will be devoted to material collected and submitted by SAGE. For the remainder of 1992, USENIX will fund up to eight additional pages of material per issue, assuming that the material is forthcoming. Furthermore, should the amount of material grow considerably, USENIX will plan to establish a SAGE newsletter in the future.

10. That the USENIX Association recognizes that financial issues have the potential to become a major block to easy cooperation with SAGE, and that the key to avoiding these problems is openness in our financial dealings and mutual striving for a fair allocation of resources. For the remainder of 1992, USENIX will pay for the SAGE election, board expenses, legal and liability expenses (if any), the cost of administering SAGE membership, and additional *login:* pages. In turn, USENIX will collect the SAGE interim dues. The SAGE board will submit a proposed budget for the fiscal 1993 year to the USENIX Board at their fall meeting, including a proposal for 1993 SAGE dues and membership estimates. The final budget will be established by the USENIX board in consultation with those members of the SAGE board so designated. We expect at that time that designated members of the SAGE board will have signature authority over the budgeted amounts.

11. That a committee be struck consisting of the Presidents and Treasurers of both USENIX and SAGE and the USENIX Executive Director to refine the SAGE Bylaws and the USENIX Special Technical Group Policy to bring them into agreement.

12. The USENIX Board will conduct a progress review in September of 1993.

Kolstad was delegated to be the USENIX representative at the first SAGE meeting, and Christiansen would be the longer term representative.

It was also agreed to allocate \$6,000 in the budget for the SAGE STG for fiscal year 1992.

SAGE Mission Statement and Charter

The Systems Administrators' Guild (SAGE) was founded with the following proposed charter, drafted by the original SAGE committee during the months prior to the public announcement by USENIX. This charter will be reviewed and modified by the incoming board as they deem it necessary.

The purpose of the Systems Administrators' Guild (SAGE) will be to advance the status of systems administration as a profession. SAGE will do so by recruiting talented individuals to the profession, by developing guidelines for the education of members of the profession, by establishing standards of professional excellence and providing recognition for those who attain them, and by promoting work that advances the state of the art and propagates knowledge of good practice in the profession.

SAGE will seek to meet these objectives by offering its members the following services:

1. Conferences

SAGE will sponsor technical conferences and workshops.

2. Publications and Distributions

SAGE will publish a technical journal and a newsletter. It will also make existing freely distributable systems administration tools available to its members, and commission the development of new tools.

3. Education

SAGE will develop curriculum recommendations for educational programs in systems administration. It will also develop recommendations and accreditation for internship/residency training programs for systems administrators.

4. Certification

SAGE will develop a process for the certification of professional systems administrators by examination, and will maintain a registry of certified professional systems administrators.

5. Job Descriptions

SAGE will develop model job descriptions for various classes of professional systems administrators.

6. Awards

SAGE will recognize outstanding achievement in professional systems administration through awards like the ACM Turing and Grace Murray Hopper Awards.

7. Public Relations

SAGE will speak for the concerns of its members to the media. It will make public statements on issues related to systems administration.

8. Local Groups

SAGE will promote the creation of local groups of professional systems administrators, like BayLISA.

SAGE Governing Board and Elections

In order to ensure a smooth transition from the design committee to the first elected governing board, USENIX has approved the appointments of the members of the committee as the interim governing board of SAGE. The members of this board are:

Shoshana Abrass, *Pacific Data Imaging*
Tina Darmohray, *Lawrence Livermore National Labs*, Secretary
Arnold de Leon, *Synopsys*
Laura de Leon, *Hewlett Packard*
John F. Detke, *Octel*, Treasurer
Paul Evans, *Maspar*
Bryan McDonald, *SRI International*, Publications Coordinator
Paul M. Moriarty, *Cisco Systems*
Arch Mott, *Protocol Engines*
Bjorn Satdeva, */sys/admin*
Steve Simmons, *Inland Sea*
Pat Wilson, *Dartmouth College*
Elizabeth Zwicky, *SRI International*, President

This board will serve until the first fully elected board takes office in January, 1993, with their first meeting being held at the Winter USENIX conference in San Diego, California. Current estimates indicate that the new board will have 4 meetings a year, three at LISA and USENIX technical conferences, and a spring meeting. At this time the board members are responsible for travel and lodging expenses for the meeting. This board will be tasked with further defining the charter and

mission statements of the guild, setting the 1994 budget, and monitoring, coordinating, and approving all the working groups efforts. Members of the board should expect to be spending significant time every week in the pursuit of these goals during the first year serving the board, after which time requirements should begin to ease a little. Since the board will be geographically diverse and working on so many issues, electronic mail will probably serve as a major conduit of information. This is a major time and financial commitment for at least the first year.

SAGE is accepting nominations for new members of the governing board until October 22 at noon, PST. Anyone interested in running for the SAGE board should send their name and telephone number and a brief statement to the nominating committee at the following email address: [<sage-nominations@usenix.org>](mailto:sage-nominations@usenix.org). You can also send U.S. Mail to the SAGE Nominating Committee care of the USENIX Association. The nominating committee will gather the candidates' names and contact each of them before the election takes place.

At the USENIX LISA conference in October there will be a candidates' forum to enable prospective board members to introduce themselves and talk about the issues. Prospective board members unable to make it to the LISA conference will be able to submit a position paper to this forum.

Report on First SAGE Meeting

by Paul Evans

MasPar Computer Corporation

SAGE held an organizational meeting at the Marriott Rivercenter Hotel in San Antonio, Texas on June 13, 1992. The meeting followed the USENIX summer technical conference, at which USENIX launched SAGE as its first Special Technical Group (STG). Forty-three participants attended the meeting. Rob Kolstad represented the USENIX board of directors for this meeting; other past and current USENIX board members were also present: Evi Nemeth, Tom Christiansen, Eric Allman, and Ed Gould. Tom Christiansen will be the long-term USENIX Board liaison to SAGE after this event.

Steve Simmons started the meeting and introduced Bjorn Satdeva, who briefly discussed the origins of SAGE, and the name – the ‘e’ that was left off the creat(2) system call has reappeared as the ‘e’ at the end of SAGE. Rob Kolstad discussed the relationship between SAGE and USENIX, emphasizing the enthusiastic support of the USENIX board for the Special Technical Groups (STGs) in general and SAGE in particular. The members of the interim SAGE governing board who were present were introduced: Shoshana Abrass, John Detke, Paul Evans, Paul Moriarty, Bjorn Satdeva, Steve Simmons, Pat Wilson, and Elizabeth Zwicky. Paul Evans read the proposed SAGE charter.

After the reading of the charter, there was brief discussion of awards which resulted in an informal decision to give “Robbies” to outstanding systems administrators, in honor of Rob Kolstad. The awards working group is currently looking into securing a supply of chubby statuettes with tutus. On a more serious note, Peg Schafer of BBN raised the concern that the proposed charter did not sufficiently take the needs of part-time systems administrators into account. It was decided to create a working group to look at issues affecting part-time administrators, and Peg volunteered to be its coordinator.

Dave Coleman of IBM asked who the intended audience for SAGE was. He felt that since PC, Mac, and bulletin board administrators outnumbered UNIX and workstation administrators by an order of magnitude, they might overwhelm SAGE. No consensus emerged on how serious this problem was likely to be, so the issue was

deferred to the working group concerned with administrators of non-UNIX systems. Dave suggested that the committee should consider not only what to do about this issue, but also whether or not to do anything at all.

Although the original intention had been to cover organizational issues quickly and move on to the formation of the working groups, the remainder of the meeting was dominated by discussion about the structure of the SAGE board. Steve Simmons presented the initial proposal for a nine-member board, which included a president-elect, president and past president, each serving two-year terms; and six directors, each serving three-year terms.

Steve proposed adopting an Australian balloting scheme under which candidates are ranked by preference. This would allow SAGE to avoid what is perceived as a major problem with USENIX elections, that unsuccessful candidates for the officer positions fall off the board altogether. However, most of the participants believed that this would be too complicated and confusing for voters.

Pat Parseghian of AT&T was concerned that paying travel costs for a nine-member board would be prohibitively expensive, but it was pointed out that SAGE has a volunteer board, and board members will be paying their own travel expenses. She was also concerned that there wasn't enough to do to justify a nine person board, especially considering that USENIX only has eight on its board.

Many of those present expressed concern about the proposed presidential succession scheme, under which one person would in turn serve two years as president-elect, president, and past president, for a total of six years. Several alternative arrangements were proposed, and after much discussion, Steve conducted a series of straw polls, with the following results:

1. The SAGE board will have seven members.
2. The SAGE board will choose its own officers from among the board members after each election (elections take place every year).
3. Board members will serve two year terms.
4. Board members are limited to four consecutive terms.

5. In the first election, three board members will be chosen for two-year terms and three for one-year terms. In deference to the strong desire of the USENIX board for continuity, it was agreed that Elizabeth Zwicky, the interim president, would automatically get the remaining one-year term on the first elected board. In subsequent elections, alternately three and four members will be chosen for two-year terms.

6. The SAGE board will determine the number and duties of its officers, within the limits of the USENIX guidelines for STGs.

7. The first election will be held after the LISA VI conference in October, on a schedule to be determined by the USENIX Association staff.

8. Nominations for the SAGE board will close at 12:00 Noon (PDT) on Thursday, October 22, to enable the candidates to present themselves at a candidates' forum at the LISA conference. Candidates who are not able to attend the

LISA conference can submit position papers to be read at the forum.

With the conclusion of the organizational discussion, attention turned to the formation of the working groups. Some coordinators had already been designated for the working groups corresponding to the eight original charter items (conferences and workshops, publications, education, certification, job descriptions, awards, public relations and local groups). However, during the week in San Antonio, other issues emerged. Steve accepted volunteers to coordinate the working groups corresponding to these new issues (ethics, policies, vendors, standards, non-UNIX and part-time administrators). The names and email addresses of the coordinators were displayed, and names for the mailing lists were chosen. [See below for the complete list of working groups, coordinators, and mailing lists.] With this, the meeting concluded.

SAGE Working Groups

SAGE was founded to meet the needs of people who administer and manage computing systems. As the organization grows, we want to have a clear vision of just what those needs are and how they should be addressed. In that light, working groups were formed at the June SAGE meeting.

Each group is chartered to discuss a topic of interest and determine if it is a viable candidate for action on the part of SAGE. The working groups can recommend against pursuing a topic further. If you have strong feelings in either direction, you are encouraged to join a group and present your opinion. If determined to be viable, the working group will draft a proposal for the SAGE board which outlines possible methods of addressing the issue at hand.

Below is the list of working groups current as of June 25. Listed are the name of the group, the leader, the email address that the working group will be using, and then a paragraph describing some of the topics and goals the group will attempt to address. If you are interested in joining one or more of these groups, send email to [`<listserv@usenix.org>`](mailto:<listserv@usenix.org>) with a body message of "help" and you will be sent further information to subscribe to the various lists.

1. Conferences and Workshops – Steve Simmons **[`sage-conf@usenix.org`](mailto:sage-conf@usenix.org)**

The conferences and workshops group will focus on public activities of both academic and practical use for systems administration, including involvement in the USENIX LISA conference.

2. Publications – Bryan McDonald **[`sage-pubs@usenix.org`](mailto:sage-pubs@usenix.org)**

The publications group is chartered to put together a series of proposals related to the various publications that SAGE wants established. In the immediate future the pubs group will be asked to assist in the publication of the first issues of newsletter segments within this newsletter. Long term goals include proposals concerning an independent newsletter, a technical journal, software tool collections, and any other ideas the committee can collect.

3. Electronic Information Distribution – Mark Verber **[`sage-online@usenix.org`](mailto:sage-online@usenix.org)**

The electronic information distribution working group will identify existing information sources that would be of use to SAGE members, new types of information that should be gathered, produced, and make proposals for the effective distribution of this information. Existing sources should include reprints of papers/articles and mailing-list/USENET news archives. New infor-

mation sources might include specially written technical/positional papers and custom databases such as vendor neutral lists of bugs. Distribution methods will include WAIS or other information servers, anonymous ftp/uucp, and CD-ROM.

4. Education – Pat Wilson – sage-edu@usenix.org

The education working group's initial goal is to outline possible plans for institutional and continuing education of the community through the development of model curricula, the identification and promotion of useful tutorial programs, and the construction of guides for self-study.

5. Certification – Paul Moriarty sage-certify@usenix.org

The purpose of the certification working group is to address the issues surrounding certification and discuss the various approaches that might be taken. The working group will present the model along with their recommendations to the board of directors.

6. Job Descriptions – Tina Darmohray sage-jobs@usenix.org

The job description group will evaluate SAGE's role in assisting system administrators with defining job descriptions. If it is determined that this is an area that SAGE should pursue, the focus of the group will be to create multiple system administration job description suites that can be used as templates for those who are writing position descriptions for hiring purposes at their own site.

7. Awards – John Detke – sage-robberies@usenix.org

The awards working group will focus how SAGE can best go about recognizing outstanding system administrators, and their achievements. Initial suggestions include a "Best of LISA" award and an annual outstanding sysadmin award.

8. Public Relations – Paul Evans sage-pr@usenix.org

As a professional society, SAGE has the opportunity to speak out to many factions of industry and/or government on issues that affect us and our profession. This committee will examine the issues, set guidelines for SAGE's involvement in this area, and (if appropriate) determine a plan or focus for the guild to evaluate and pursue (or not pursue). This group will also serve to direct any future public relations issues that may arise.

9. Local Groups – Bjorn Satdeva sage-locals@usenix.org

This group will have the task of exploring how

the creation of new local groups can be made easier, how SAGE and USENIX can assist in their formation, and how SAGE can support existing local groups.

10. Ethics – Ed Gould sage-ethics@usenix.org

This group is charged with determining SAGE's role in developing a set of guidelines or codes of ethics for the system administrator. We see these guidelines or codes as having at least two purposes, namely, guiding one's self in the performance of system administration tasks, and informing one's employer and co-workers of the proper bounds of system administration.

11. Policies – Lee Damon sage-policies@usenix.org

The focus of the policies working group is the consideration of one or several documents for system/network administrators to use as guidelines or boilerplates in setting up their site's policies and procedures. We will also be working with the education and ethics working groups to help systems administrators understand just what policies are, and why they can be important.

12. Vendors – Terry Bartlett sage-vendors@usenix.org

The vendor working group will evaluate SAGE's role in establishing a consensus on the types of tools we'd like vendors to provide for system and network administration. The group may also act as a vendor lobbying organization to convince the various manufacturers to "do the right thing" when it comes to system administration.

13. Standards – Janet Frazer sage-stds@usenix.org

The standards working group will evaluate the potential for SAGE to monitor and affect the various standards bodies that are currently or will be in the future setting standards for system administration.

14. Non-UNIX – John Detke sage-outreach@usenix.org

The non-UNIX outreach working group will focus on how SAGE can remain pertinent to people who manage computing systems and networks, addressing the issues that cross operating and network system boundaries.

15. Part-Time Admins – Peg Schafer sage-pt@usenix.org

The part-time working group is concerned that the proposed SAGE charter did not mention the large number of people who do system adminis-

tration less than 100% of the time (e.g., there are chemists who fulfill system administration roles some portion of each working day; yet, they view themselves as chemists, not system administrators). We will consider how SAGE could address the needs of such individuals and propose changes to the proposed SAGE charter which address this "dual-role" reality.

Security is a growing concern for system administrators. This group will examine ways of helping system administrators assess their need for security and security policies. We will seek ways to educate system administrators on security issues. We will consider soliciting or developing sample policies and tools in this arena.

SAGE Book Reviews

by Steve Simmons

[<scs@lokkur.dexter.mi.us>](mailto:scs@lokkur.dexter.mi.us)

What the world needs is a good book on the hardware side of ethernet – installing, expanding, maintaining, and debugging. Unfortunately there is no such beast. This review will discuss two publications which cover the physical side of ethernet.

***Keeping The Link* by Martin Nemzow. McGraw-Hill, 1988, ISBN 0-07-046302. 366 pages. Hardcover.**

This book covers the physical end of various flavors of ethernet. It contains a great deal of good material, as well as some non-technical material which can safely be ignored. It has some deficiencies, but is quite useful in spite of them.

It is excellent at covering the purely physical and technical topics. Its most valuable feature is the detailed treatment on the physical handling of an ethernet. It includes step by step instructions with photographs and drawings on a number of topics, including: how to make taps; how to debug physical and electrical problems using TDRs and various other test equipment; (the section on TDRs includes photographs showing the traces from various sorts of ethernet hardware in both proper and defective operation); and drawings and pictures of various common cables and other connection hardware.

From this book, I was able to correctly install a thick ethernet transceiver, having never seen the tools before.

In addition to the excellent instructions, it is rich with diagrams, charts, and tables of physical constants. They're often worth as much as the text. It also contains a number of sample forms

and recommendations for managing the physical cable plant. These should all be of great use to any working administrator.

Unfortunately, it has a number of problems. Nemzow is a firm believer in broadband ethernet and gives it equal play with thicknet, Cheapernet, and thin ethernet. He discusses fiber, but at a much lower level of detail; 10baseT is almost completely ignored. Given that the book was written in 1988, these last two points are somewhat forgivable, but the lack of data on 10baseT lengths is particularly frustrating.

Nemzow spends a great deal of time talking about the usefulness of networks. This material is not needed, appropriate, or accurate. Fortunately it's easy to skip over.

In summary, this is easily the best of what I've seen on the hardware-side of ethernet management. It is not a great book, but nonetheless is a valuable addition to your library. A second edition with updates could be a major seller.

***Telecommunication Wiring* by Clyde N. Herrick and C. Lee McKim. Prentice-Hall, 1992, ISBN 0-13-151531-4. 253 pages. Hardcover.**

I ordered this on the basis of a flier from Prentice-Hall which touted it for the physical end of computer network management. The back cover reiterates this claim. Unfortunately, the contents do not live up to the claims.

This book has major flaws for anyone using it as a guide for computer network installation. It repeatedly mentions using coaxial wiring for cable TV, mentions that ethernet runs over coaxial cable, but never mentions that the two require different sorts of cable.

Similar problems can be found with the telephony wiring sections. No mention is made that one might want to wire telephony systems somewhat differently from 10baseT or RS-232.

In short, this is a most disappointing book for the computer network management and I do not recommend it.

SAGE Membership Form

Yes, I would like to join the USENIX special technical group SAGE, the Systems Administrators' Guild, as follows:

- [] I am a current USENIX member. I wish to join SAGE. Enclosed is \$25 to cover dues for the remainder of 1992. My membership number is _____.
- [] I am not a current USENIX member. I wish to join USENIX and SAGE. Enclosed is \$90 (\$65 for a one year individual membership in USENIX; \$25 for SAGE dues for 1992).

Name _____



Address _____

City _____ State _____ Zip _____ Country _____

Phone _____ email address: _____

PAYMENT OPTIONS

☐ Check enclosed payable to USENIX Association or SAGE.

☐ Please charge my: ☐ Visa ☐ MasterCard  

Account # _____ Exp. Date _____

Signature _____

Outside the U.S.A.? Please make your payment in U.S. currency by one of the following:

- * Charge (Visa, MasterCard, or foreign equivalent)
- * International postal money order
- * Check - issued by a local branch of a U.S. Bank

USENIX Mailing List

- ☐ I do not want my address made available to other members.
- ☐ I do not want my address made available for commercial mailings.

Please mail this form to: USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

Simmons' Laws of System Administration

by Steve Simmons

<scs@lokkur.dexter.mi.us>

The Definition:

System Administration is the combination of system support and user support.

The First Law of System Administration:

Any rule can be modified by the application of power and policy. By contrast, rules always are subordinate to laws.

The Network Paradox:

System support is a subset of network support. Network support is a subset of system support.

The Laws Of Unanticipated Support Cost:

1. It will always cost you more to support a thing than the vendor told you.
2. It will usually cost you more to support a thing than to buy it.
3. Sometimes it costs 10x as much to support a thing as it did to buy it.
4. Refusing to support something often results in the thing being unusable.
5. Once it's installed, supporting a thing is sometimes cheaper than not supporting it.
6. Before buying, make sure you're committed to support. But see item 1.

The Division Between System Support and User Support:

There's a difference between system support and user support. There may be overlap in the two positions; sometimes both are done by the same person. But the two tasks are distinct and sometimes have conflicting goals.

The Law Of Distributed Talent:

Great system support people often make lousy user support people and vice versa.

The Paradox Of Dual Abilities:

The person good enough to do both system support and user support will usually be hired away by a shop where the combined tasks are too large for a single person.

On Complexity And Customization:

Application-to-application differences confuse everyone, especially users and support staff. Ditto UNIX-to-UNIX differences, etc. By contrast, complete consistency completely stifles improvement. At any given site for any given application or feature, there's someone who knows more about it than the support staff. Finding that person is the first step to take to diagnose any given problem.

Time to diagnose and time to fix are completely unrelated. Sometimes one approaches zero while the other approaches infinity. This is especially hard to deal with when the diagnostic person and the fix person are not the same.

One person's improved feature is another person's gratuitous change.

Users want applications and systems they can customize.

One user's customization is another user's gratuitous change.

The Laws Of The Cost Of Customization:

The cost of customization is complexity. The cost of complexity is increased difficulty in administration and user support. The cost of increased difficulty in administration and user support is either lower quality of administration and user support, increased support staff, or both. Therefore, increased customization means increased cost, or lower quality of support, or both.

The Paradox Of Unused Customization:

It doesn't matter whether customization has actually been done. The mere fact that it's possible means you must check for it, thereby increasing the cost of problem diagnosis.

Smallwood's Law (Simmons' paraphrase):

"They're not users, they're clients." Kevin Smallwood.

Users Are Human:

The user who says "Can X be done?" is usually really asking "Would someone please do X?" Make sure you answer both questions. It's human

to blame problems on outside causes. By contrast, an outsider will always suspect the insider as the cause.

The user who says "I didn't change anything" isn't always lying. Sometimes they're just ignorant or forgetful.

It's more important for users to do their job than to answer the needs of admins. Unless of course their job is to answer that need.

Admins Are Human:

For every statement in "Users Are Human," change "user" to "admin" and vice-versa.

The 'You Broke It' Principle:

Cockpit error is the most common cause of problems. Everybody is a pilot.

Support Is Overhead:

One way of cutting costs without cutting development staff is by cutting overhead. System administration and user support are overhead.

User and system admin training are overhead. Not having them increases overhead. Go figure.

The Joy Of Being A Contract System Administrator:

"Sure, we can do that. Here's what it'll cost you."

His Site Isn't Your Site:

The situation at your site doesn't make you qualified to judge the situation at another site, and vice-versa.

Just because someone else's support staff does it doesn't mean your staff can do it. (This statement is subtler than it looks.)

The Rules of Policy and Power:

1. System administration is whatever the boss tells the admins it is.
2. Users will bypass admins to get the boss to tell the admins something different. That's their right.
3. Most system admins live in a policy vacuum. This can be good or bad:

Corollary 1:

Power expands to fill a vacuum. That thing which expands most easily is a gas.

Corollary 2:

Anything that quickly expanded to fill a vacuum is easily displaced by a solid.

Corollary 3:

A rapidly moving solid will hurt you if you're in its way. The person who does your job review makes the rules. The good admins always follow those rules. See Rule 1 and the First Law.

The Summary:

Be careful what you do in that vacuum. Nobody appointed you God. However, you can always be disappointed.

The Laws Of System And Network Growth:

You can always incrementally add one more.

Sometimes the straw breaks the camels back. More often, the camel just goes slower and slower.

The difficulty of support does not grow linearly with the size of the site.

Eventually your site outstrips your methods, and you must bite the bullet and move to new methods.

Corollary:

Nobody bites the bullet until there's not enough time to do the existing work. At that point there's not enough time to make the changes.

Adding a new kind of computer, operating system, application, peripheral, etc., has a much higher administrative cost than adding one more of what you've already got.

Corollary 1:

If you buy one, you may as well buy ten.

Corollary 2:

If you buy ten, you may as well buy eleven and keep one for spare parts.

Name Frequency on Usenet

by Elizabeth Zwicky

<zwicky@erg.sri.com>

Inspired by the ZONE (Zealot of Name Edification) data about host name frequency in the DNS databases, I wrote a Perl program called "names" that scavenges /usr/spool/news for host and user names. The program attempts to get as many names as possible, by looking for addresses in the entire article. It does apply various filters; for instance, it recognizes Path: headers and throws out the last component (which is sometimes a user, sometimes a host, and sometimes a place filler like "dont-send-mail-to-path-lines"). It also attempts to recognize encoded binaries (and ignore them) and article IDs (from which it rejects the "user" and saves the "host"). It stores the user-host pair, so that people who post multiple times from the same host are not multiply counted.

Because of the catch-all approach, the names program has a tendency to undo the efforts of sites to hide individual host names and present only site names. This is beneficial to the host name data, but confounds the user name data by turning a single user who posts from multiple machines at the same site into multiple users – for undergraduates who post a large volume of news from many different machines, this rapidly becomes a serious problem. This is filtered by a program which attempts to condense hosts that a given user name posts from. Some duplicates remain, mostly from single hosts on multiple networks, and a few hosts are incorrectly condensed, but the result appears to be considerably more accurate than the uncondensed data. For a user like me who appears at "pterodactyl.erg.sri.com", "sparkyfs.erg.sri.com", "erg.sri.com", and "sparkyfs", it will condense the whole lot down to "erg.sri.com".

The host name counts just look at each unique host name. This will result in some duplication for hosts that appear as "host", "host.com" and "host.uucp". The program's trusting nature also causes it to take example addresses as valid, which probably explains the popularity of "host" as a host name. (It does not fully explain why "nodomain" is a more popular domain than "domain", however.)

An option to the program allows it to accumulate data over time; I have spent several days letting it

run every night out of cron to add new names. This uses disk space, memory, and CPU cycles more profligately than many people may wish to use them, but helps to separate the data from the noise. It also amuses me to watch the numbers move around with added data. After 10 days, I am averaging about 4,000 new users and 1,000 new hosts a day. Still, I have less than a tenth as many hosts as the DNS data. These figures cover news in the spool directories on news.erg.sri.com between January 21 and February 10 1992. For those who wish to play along at home, the Perl scripts I used are available for anonymous FTP on ftp.erg.sri.com. They are both CPU and memory intensive, and consist primarily of uncommented Perl code with regular expressions that make Tom Christiansen's head swim. Let the FTPer beware.

The most popular names are listed. There are several thousand way ties for least popular name (of course, truly unpopular names never occur at all, and there are millions of those), but I have listed the ones that struck me as notable.

User Names

Out of 114,195 users, there were 59,369 distinct user names. The top 50 user names:

Rank	#	Name
1	4603	postmaster
2	1628	root
3	1382	news
4	648	usenet
5	191	steve
6	186	mike
7	179	mark
8	170	john
9	166	dave
10	148	paul
11	142	chris
		david
13	133	bill
14	125	brian
15	124	peter
16	123	bob
		jim
18	118	scott
19	113	tom
20	112	user
21	106	dan
22	104	jeff
23	98	jim
24	97	eric
25	93	rick
26	88	listserv
27	84	ken
		larry

Rank	#	Name (cont.)	USENET Rank	ZONE Rank	Name (cont.)
29	83	alan	10		sun
30	81	joe	11	(12)	orion
		martin	12	(37)	athena
		michael		(20)	sirius
33	80	kevin	14		vm
		system	15		physics
35	75	marc	16	(17)	hobbes
36	73	george			oak
		jon	18		eagle
		robert	19		helios
39	72	greg		(4)	jupiter
40	71	rob		(10)	mercury
41	69	andrew		(36)	phoenix
		andy			ra
		bruce	24		astro
		doug			rigel
		keith		(25)	thor
46	66	info	27		chaos
47	65	frank			informatik
48	64	gary		(3)	mars
49	61	richard			relay
		ron	31	(46)	gandalf
				(28)	titan
			33	(44)	earth
				(17)	neptune
			36		vax1
				(14)	csd
					gauss
					odin
				(1)	turing
			41	(27)	venus
					calvin
					edu
					eng
					host
					isis
					nova
			47		hydra
					lynx
					mail
				(26)	merlin
				(25)	opus
				(7)	saturn
					sun1
			54	(32)	apollo
			64	(48)	alpha
				(15)	newton
			83	(30)	hermes
				(2)	pluto
			99	(35)	hal
				(6)	iris
				(39)	mozart
			148	(31)	fred
				(33)	grumpy
				(19)	gw
			182	(47)	io
				(40)	snoopy

The list continues similarly, with male first names and system accounts, until rank 94, which introduces the last names "anderson", "brown", and "johnson". At rank 133, the first exclusively female first name, "karen", appears. (It is of course impossible to tell how many of the instances of "pat", "lee", and so on actually belong to women, and how many are last names. "lee" and "kim" in particular may be first names of either men or women and are also frequent last names.)

Host Names

Host names were divided up into first components, middle components, and last components. (A name with only one component was considered to be a lonely first component.)

Out of 56,340 host names, there were 33,305 distinct first components. These are the top 50. The number in parentheses is the rank from the ZONE information about DNS; all names in either top 50 are shown.

USENET Rank	ZONE Rank	Name
1	(5)	cs
2		news
3		math
4		cc
5		ee
6		p0
7	(45)	sol
8)	(9)	zeus
9		vax

(Names with fewer than three occurrences are not ranked. The extra number in the following items is number of occurrences.)

(8)	0	pc1
(11)	0	mac2
(13)	0	mac1
(18)	1	pc2
(21)	0	mac3
(24)	0	cisco
(29)	0	mac4
(34)	0	mac5
(38)	0	mac6
(41)	0	pc3
(42)	0	mac7
(43)	0	mac10
(49)	0	mac9
(50)	0	mac8

Some of the differences between the USENET data and the DNS data have obvious causes; not many people on Macs or PCs post news, which explains why names like "mac1" and "pc1" almost never appear in the USENET data. The name "p0" turns out to be very frequent in Fidonet addresses. Some of the remaining differences may also be minimized as the amount of USENET data grows. Several of the top 10 names from the DNS data ("saturn" and "venus", for instance) did not initially make the USENET top 50, and have been climbing steadily. "snoopy", on the other hand, appears to be destined to remain an also-ran. "elvis" is steadily climbing, and currently at rank 73.

A randomly chosen selection of the least popular machine names (there are also a few machines named after phone numbers):

aethelbehrt
air-traffic-controller
anywhere
apple-gunkies
bad-loud-music
biscuit-tin
bopbopaloopbopawhopbamboom
broccoli
brussel-sprout
computername
couqsmungus
deathstar
excaliber
fnord
globus-pallidus
grubby-thicket
his-phoenix-multics
kibblesnbits
killer-tomato
moo
moving-target

national-institute-for-medical-research
pickled-brain
poseur
ratmandu
rent-a-guru
schoolofthought
snorkelwacker
squid-lips
studguppy
toaster
tofu-hut
welsh_git

Middle Components

Everything between the first and last components was lumped together as a middle. Middle components have less individuality than first components, with a mere 5,788 distinct middles out of a total of 49,731. "z1" is another artifact of fidonet.

Rank	Number	Name
1	1456	cs
2	1250	ac
3	897	co
4	695	dec
5	684	oz
6	634	sun
7	524	nasa
8	489	hp
9	84	edu
10	473	fidonet
11	471	cc
12	464	eng
13	450	berkeley
14	427	enet
15	372	z1
16	343	mit
17	298	att
18	288	ibm
19	284	stanford
20	283	sgi
21	268	cis
22	266	ohio-state
23	249	uiuc
24	229	cmu
25	211	informatik
26	208	umich
27	200	wisc
28	192	msk
29	177	ge
30	176	purdue

Domain Names

The domain name figures are consistent; although new domains appear as more data accumulates, the rank orderings have been nearly stable since the first run. The second number is the rank for the domain in the ZONE statistics takes

from DNS. All domains in either top 30 are shown. The Soviet Union still appears in these lists, despite the fact that it ceased to exist nearly a month before I began gathering statistics. This is because there are still a significant number of hosts in the .su domain while naming settles out.

USENET Rank	ZONE Rank	Name	Country
1	(1)	edu	
2	(2)	com	
3		uucp	
4	(7)	ca	Canada
5	(8)	org	
6	(15)	uk	United Kingdom
7	(5)	au	Australia
8	(3)	gov	
9	(6)	de	Germany
10		bitnet	
11	(16)	jp	Japan
12	(9)	se	Sweden
13	(14)	nl	The Netherlands
14	(11)	fr	France
15		su	Soviet Union
16	(4)	mil	
17	(12)	fi	Finland
18	(25)	us	United States
19	(13)	no	Norway
20	(10)	ch	Switzerland
21	(21)	dk	Denmark
22	(22)	nz	New Zealand
23	(23)	es	Spain
24	(17)	at	Austria
	(18)	net	
26	(19)	it	Italy
27	(26)	be	Belgium
	(20)	il	Israel
29		oz	Australia
30		deenet	
36	(29)	is	Iceland
46	(27)	mx	Mexico
48	(30)	br	Brazil
52	(24)	kr	Korea
55	(28)	gr	Greece

Comparing countries represented in the USENET data to Don Wells' data on countries connected to the Internet as of January 28, 1992:

Continent	Both	Internet	USENET
Africa	Tunisia South Africa		Zimbabwe
Antarctica ¹		Antarctica	
Asia	Hong Kong		China

1. The Antarctic belongs to no country, by treaty. The USENET data is based on domain extension, and since Antarctica does not have its own domain, it is invisible.

Continent	Both	Internet	USENET
	Israel India Japan Korea Singapore Taiwan		Sri Lanka Nepal? Malaysia
Australasia	Australia New Zealand		
Europe	Austria Belgium Belorus Czechoslovakia Germany Denmark Finland France Greece Hungary Ireland Iceland Italy Latvia Netherlands Norway Poland Portugal Spain Soviet Union Sweden Switzerland Ukraine United Kingdom Yugoslavia		
N. Amer.	Canada Mexico United States		
S. Amer.	Argentina Brazil Chile	Paraguay Cuba? Costa Rica	
Other			American Samoa? Netherland Antilles?

Countries marked with question marks have hosts that I cannot identify as mistakes, but that are not immediately identifiable from their names as being bona fide, either. Errors and non-Internet addresses are both common in the USENET data; I checked the domains by hand for plausibility and eliminated several dozen apparent countries that way. Colombia's surprisingly large presence on USENET turned out to consist entirely of .com sites with the "m" chopped off, for instance.

An Update of UNIX-Related Standards Activities

by Stephen Walli

Report Editor <stephe@mks.com>

USENIX Standards Watchdog Committee

You are in a Maze of Twisty Profiles — All Different

[Warning — Profiles are poorly understood, ill-defined specifications that are being drafted as full standards in various corners of the standards community. If at first the article seems twisty and convoluted, that is because the topic is twisty and convoluted, mired in a lot of historical context. The article presents the historical context for profiling activities, and the traps lying in wait for unsuspecting applications developers. It finishes with a few recommendations.]

Profiles are the latest confusion to appear on the open systems standards scene. They are supposed to define a view on one or more standards in a coherent way to fulfill a general need. This need may be something like: "a programming platform for general multi-user, multi-processing business applications" or maybe: "supercomputing applications typically require the following services." This seems reasonable. It also seems to feel right. So what happened?

In the Beginning ...

The POSIX.1 (ISO/IEC 9945-1:1990 == IEEE Std. 1003.1-1990) standard standing alone is not enough. By its own definition, it requires C language support. This can be either Common Usage C or Standard C (ISO/IEC 9899:1989 == ANSI X3.159-1989). These two standards together provide a reasonable programming environment. They are not complete; there are many things missing. To move the standard forward, things were left out that were too contentious at the time. It was better to have some kind of standard than none at all.

POSIX.1 also has optional functionality. Some of this functionality is called out by "Big-O" options, such as `{NGROUPS_MAX}`, `{_POSIX_JOB_CONTROL}`, or `{_POSIX_CHOWN_RESTRICTED}`. These are implementation level options, and a vendor could choose not to implement them and still be conforming. There are other named options, such as `{_POSIX_NO_TRUNC}`, and `{_POSIX_SAVED_IDS}`, which may or may not be implemented. A strictly

conforming application should never count on such functionality being present.

Using this simple model, the National Institute of Standards and Technology (NIST) created the U.S. government procurement document, FIPS PUB 151-1. In it, NIST specified what options and limits must be supported from POSIX.1, and how the C language support should be done. The intent was to provide as functional a platform as possible by mandating as much of the POSIX.1 standard as possible, something upon which U.S. government applications developers could depend. Simple.

A long time ago, relatively speaking, X/Open was formed. It described a collection of specifications that all of its member vendor organizations would adhere to. Thereby it provided a Common Application Environment (CAE) for applications portability. This specification of a platform's functionality was written down in the X/Open Portability Guide (XPG). They have made a point of adopting POSIX standard interfaces where possible, moving away from the original SVID definitions. Perhaps not complete, but still relatively simple. Both FIPS and the X/Open XPG feel kind of like something you, as an applications developer, might want to point to when describing the environment you want.

Now let's move on to where things start getting messy. A number of things start happening in parallel, which means the confusion factor goes up exponentially. POSIX began doing some things. ISO was doing others. The industry consortia were doing something else. And remember, the industry consortia are the ones backed by vendor money, and have a stake in selling you their solution. Industry consortia == A vendor once removed.

POSIX

A few years ago, at the beginning of the Great Project Proliferation in POSIX, two projects began which would develop Applications Environment Profiles (AEPs) for Supercomputing (POSIX.10) and Transaction Processing (POSIX.11). The intent was to describe how to use POSIX in building applications in these two particular domains. In the last two years, two more AEP projects developed in POSIX, one for Real-time applications (POSIX.13) and one for Multi-processor applications (POSIX.14).

These last two are illustrative of many of the problems encountered. The POSIX.4 (Real-time) and POSIX.4a (Threads) standards will become addendums to the POSIX.1 base standard. All function interfaces defined by POSIX.4 and POSIX.4a will need to be provided by future implementations of POSIX.1, although they may be just stubs returning ENOSUPPORT (or some such), if the implementation does not support the added functionality. This functionality will be called out by named options. Hmmmm. Getting a little muddy.

POSIX.13 and POSIX.14 would hopefully define an applications domain that must be provided by an implementation for the appropriate class of applications. By pointing at the appropriate base standards and choosing options, we can clearly define the requirements of a class of real-time or multi-processing applications. It is unclear whether the base standards are POSIX.1, POSIX.4, and POSIX.4a, or some future, as yet completed ISO/IEC 9945-1:2001.

That's perplexing enough. Now consider the following. POSIX.6 (Security), POSIX.8 (Transparent File Access), POSIX.12 (Protocol Independent Interfaces), POSIX.15 (Batch), and POSIX.17 (Directory Services) functions will all be grafted onto POSIX.1, with options, as they are approved. All of these base API standards, some of which are nothing more than option-labelled "diffs" to POSIX.1 (i.e., POSIX.8), will somehow be fit together into one BIG book. (And people thought POSIX.2 was big!)

Remember, all of the function interfaces will need to be provided by an implementation, even if only as stubs because the "option" is not provided by the implementation. A portable application will spend all of its start-up time querying `sysconf()` to determine if the underlying support is present. Profiles, which strategic management believes will provide some wonderful shorthand notation to discuss procurement packages with vendors, will do nothing for the applications developers actually writing applications.

Application Environment Profiles

I made reference to AEPs defining an environment that must be provided by an implementation to support an application domain. This is another source of confusion. Are we specifying an application domain where the implementation supports far more? It likely does anyway, but in a non-standard fashion. Or are we specifying a "platform" environment, so it provides a broad base of functionality typically required by an application domain. I believe this ambiguity lies

at the heart of the "sub-setting" problem between POSIX.1 and POSIX.13.

The "sub-setting" argument arises because the real-time AEP (POSIX.13) wants the ability to call out parts of POSIX.1 as options, e.g. the file system. Some people feel this is a horrible idea, since POSIX.1 specifies a good general purpose base upon which to build applications. The profile specifiers, however, don't need the rest of the standard to describe their application domain. This has been a constant source of argument and confusion in the POSIX world. What can a profile point to, and how?

And then there are other specifications, outside of POSIX and TCOS, that would be obvious to include in certain application domain profiles. The POSIX.14 Multi-processing AEP would like to point to the X3 parallel language extensions work. The IEEE has no problem with pointing to other specifications, even incomplete early drafts such as the case here. It might even be an algorithm in a textbook.

If the point is to define a standards based environment, why would anyone want a profile standard to point to an indeterminate draft of a standard which is very unstable, even once it is mature enough to ballot? De facto specifications from vendors and vendor consortia (such as PostScript or OSF/Motif) are more stable than this!

ISO, on the other hand, has very strict rules about what can be pointed to in a profile. This leads us to another fine source of information and confusion. Let's look at ISO's contribution.

ISO and the OSI Stack

ISO has a little more experience with profiles, or maybe one should say longer experience. If I understand things correctly [salt warning]:

The ISO SC21 working groups defined the now famous seven layer stack. This was an anticipatory model, telling us how things should/would be done in the future, rather than one cluttered by implementations.

Vendors were somewhat horrified when governments started leaning in this well defined, robust direction. They still wanted to be able to play in the lucrative government sandbox. They started demonstrating how, if you interpret things in one light, their product fits the model here, or really fulfills these two layers together over there, and so on. The stack mutated a little.

The wonderful situation arose, that it was now possible to draw an entire path through the stack, top to bottom, which wouldn't communicate with another line through the stack. People even gave this a name: *conforming incompatible implementations*.

The procurement agencies weren't too thrilled by this turn of events, and profiles were born. U.S. GOSIP (Government OSI Profile) specified a known implementable path through the maze, and they used this for procurement specifications to ensure that one government OSI installation could communicate with another.

So we now have this concept of a profile. Choose a set of API and protocol specifications that will work together to form the OSI communications models. ISO even developed a document specifying how to do this. Technical Report 10000 (TR10000, or TR10K) defines a set of rules for how to define an OSI profile.

TR10K has very strict ideas about how OSI profiles are to be constructed, what they can point to and how. Profiles can only point to ISO standards (or other ISO profiles), if they are to have normative weight. Otherwise, the references are just informative.

Chaos Sets In ...

When the full complexity of this profiling problem began to appear, a number of different working groups began investigating the problem from various angles.

The profiling groups within POSIX were identifying problems as they built their drafts almost from the time they started meeting. The groups operated fairly autonomously, however, and initially never got together.

Appeals were made to the POSIX.0 working group for help. The POSIX.0 Guide to Open Systems Environments defines a model for how strategic management views standards being used, identifies many standards and where they fit into the model, and even has a couple of chapters on profiling activities and how they should be done. The POSIX.0 working group argued, however, that it was not responsible for setting profiling policy. Go figure.

After much pain and gnashing of teeth by the four POSIX profiling groups, a TCOS steering committee was formed to help solve the problems they had been having for about two years at this point. The group is made up officially of one member from each of the working groups defin-

ing profiles, and a few members of POSIX.0. Really.

The Profiling Steering Committee has been meeting for a year now. They were immediately lost in a forest of liaison points, and information gathering, trying to determine the state of profiling in the world. Now to my poor naive way of thinking, someone is not doing their job here. If the POSIX.0 members of the PSC did not already have all of the profiling documents that could be found, upon what is the profiling material in POSIX.0 based? Conversely, if they did have the profiling information and experience, then why has it taken a year to define a set of rules by which IEEE POSIX working groups should be defining profiles?

And even with a Profiling Steering Committee, they were so busy investigating what everyone else was doing, no one noticed that the POSIX.13 Real-time profiles were in ballot. Takes your breath away.

On the ISO front, things aren't much better. True to their anticipatory nature of late, a few different groups have been formed to investigate and comment upon something which doesn't yet exist. Technical Specification Group 1 (TSG1) has taken a kick at the cat.

The Special Group on Functional Specifications (SGFS) is also giving it a try. SGFS is attempting to take the TR10K document and modify it in a couple of places so as to make it applicable to the functional API standards, such as POSIX.

The European Workshop on Open Systems (EWOS), a CEN/CENELEC sponsored body, has set-up a working group to investigate a Common Application Environment (CAE). This work may be the most pertinent to date. There are people in this working group that have actually spent time attempting to specify real profiles in the commercial world. X/Open is involved in the work, lending its experience with defining specifications such as XPG3.

The EWOS work attempts to define a method of investigating the user requirements, building up the definitions and interfaces (informational rather than actual programming interfaces), and only as the very last step investigating how standards might be applied to the requirements model.

I was careful in the last paragraph to not say what type of profile was being defined. There is still a lot of discussion with respect to what is an application environment profile, versus a platform

environment profile, and there is even a new concept of a component profile. There is grey, and then there are shades of grey.

Wrap Up

```
> GET SENSE
  I SEE NO SENSE HERE.
> XYZZY
  XYZZY DOES NOT WORK HERE.
> DROP THE BIRD
```

Where do we go from here?

Profiles are poorly defined specifications (despite the many attempts at writing rules for their creation,) based for the most part on unstable documents in ballot, and there is no real experience at defining and implementing formal profiles in the open systems world. (The OSI profiles appear to be a well-defined, well-structured set of specifications which developed after there was experience with the stack — not before.)

Why do people feel that these documents should be standards? Why build castles on foundations of sand? We do NOT know what shape some of the key standards will be in until they finish ballot. Simply pointing to an interim draft of a document in ballot, even if the IEEE is willing to archive the draft, is silly.

The point is to specify an environment (application specific or not) which applications developers can count on. These environment specifications are supposed to be standards based. That's the whole point! The draft documents in ballot will change. Saying we'll modify the profile standard later, when the base documents complete ballot, is naive! People will have used it in procurements. Applications will have been written. What if the functionality in the base standard is gone? Or mutated so as to be useless to the profile? What's the rush for a useless standard?

There is the desire to specify how a set of standards can be used together to define a known environment to solve a set of applications portability problems. The simple extremes of a single standard profile (FIPS PUB 151-1), or a suite of specifications (X/Open's Portability Guide), have proven to be useful. It is useful for people to put down on paper the definition of a set of requirements for a particular applications domain, as is described by documents like the EWOS CAE working group's work. This should be done in a less formal way.

The Paul Masson Method should be applied. (We will define no standard before its time.) Make the

profiling work either "guidelines" or "recommended practices" at the IEEE level, or "technical reports" at the ISO level. Until people have REAL experience putting these complex, subtle API definitions together with appropriate other functional and language standards, many of which are still in ballot or under definition, profiles should not be given the weight of full standards.

If you're an application developer, get involved. Follow the POSIX mailings. Determine what your national standards organization is doing. Ask questions, or make yourself heard through your institutional representatives to POSIX. (USENIX, EurOpen, Uniforum, DECUS, CUG, and SHARE are all represented in IEEE TCOS. X/Open, Unix International, and the OSF are also present.)

Before some strategic thinking manager above you makes the decision for you, you should fully appreciate the enormity of the confusion being unleashed on you as you quietly contemplate that POSIX.1 and ANSI C are probably useful after all.

Report on POSIX.0: Guide to Open Systems Environments

Kevin Lewis <klewis@gucci.enet.dec.com> reports on the April 8-12, 1992 meeting in Dallas, TX:

As I reported in the January Snitch for POSIX.0, the POSIX Guide to Open Systems Environments (OSE) is going to formal ballot (finally, as someone in the SEC said to me...). If you are in the TCOS Balloting Pool, you should have received an invitation to join the ballot group for this work.

The formal ballot will be on draft 15 which is being produced presently. The changes submitted by the group to produce this draft strictly addressed the mock ballot comments. The group agreed (after I placed a gag order on them) not to surface or open any issues that had previously been considered closed. This went a long way towards our moving through the comments and objections. (By the way, if you were one of our mock balloters, please be patient. I will be sending out a summary along with detailed ballot resolutions after I have completed the formal ballot package for IEEE.)

The formal ballot closure date has not yet been determined, although it appears that the end of August is the likely time frame. Our goal is to have a significant number of ballot responses into IEEE and the ballot coordinator (i.e. me) prior to the October meeting in Europe so we can use that time for ballot resolution, as well as share the results with our European counterparts.

Two issues remain as we move toward formal ballot. One is a rationale document. It became apparent during the April meeting that our attempts to use our Issues Log, along with the minutes and institutional memory of the core participants of the group, were lacking when it came to actually documenting our rationale for certain issues. So the July meeting has been dedicated to the task of developing and writing this document.

The other issue is that of public specifications. Our document is moving into the international formal standards forums. Looking out to the horizon, we expect there will be a lot of consternation over the group's choice to include informal specifications in the guide.

This will undoubtedly shape into another battle of significant proportions. The formal ballot period should offer the current warriors a chance to take a breather.

Report on POSIX.1: System Service API

Peter Collinson <pc@hillside.co.uk> reports on the April 8-12, 1992 meeting in Dallas, TX:

[Ed. — Peter is the USENIX Institutional Representative to TCOS-SS, the IEEE group responsible for drafting the POSIX family of standards.]

Overview

Theoretically, I spent most of the week in POSIX.1, the working group for the "original" system interface standard. It's still meeting because it has several extant projects:

POSIX.1LIS, programming language independent POSIX.1;

POSIX.16, the C binding to POSIX.1LIS;

POSIX.1a, the place where bug fixes and new features for POSIX.1 are being put while the language independence work is being done;

POSIX.18, the POSIX Environment Profile. It's a profile (or list of other standards) intended to describe something close to a complete UNIX system.

I tend only to attend the work for the first of these because I also go to many other steering committee meetings. Here's an idea of what happened in the bits that I managed to get to

The Report...

The ISO standards working group on POSIX,

WG15, requires that the IEEE POSIX working groups produce a programming language independent version of the existing POSIX.1 standard (ISO 9945-1). This language independent specification (LIS) is referred to as POSIX.1LIS.

The POSIX.1 standard has been re-cast in two sections: the language independent specification and a C language binding (POSIX.16). The idea is that these two should ballot together, so that balloters can compare the original standard with the new pairing.

It's planned now that the two standards will go to ballot on July 7th. This has been made possible because:

the documents are close to being ready, have been mock balloted and finally preened by the working group

the Steering Committee on Conformance Testing (SCCT) has agreed that the documents do not need a completely new set of test methods written for them. They can use the already existing test methods for POSIX.1, contained in POSIX.3.1, which has nearly completed balloting.

Not needing new test methods is a great concession because it avoids the rule that insists on test methods being available for all new standards before they go to ballot. In my opinion, someone will need to find some funding to get the new test methods written. There is no enthusiasm for doing this in the working group. This is also the consensus of the group, when asked just that question.

What are test methods? That's a little hard to explain. Basically, they are terse English statements that assert facts about the standard. The idea is that these are easier to convert into programs that actually test the interfaces. Each assertion is classified as "testable" or "not testable," and whether or not it applies to optional behavior. It's a little more complex than this. Look at POSIX.3 (IEEE Std. 1003.3-1991), the standard for test methodologies for POSIX, for more information.

The current document drafts are based on the ordering in 9945-1. This is good because sections in all the documents refer to the same material. If you are looking at Section 3.2.1 in 9945-1:1990, then the same material will be found in the same numbered section in POSIX.1LIS and POSIX.16.

A small group of people who are close to the document — the editor (Hal Jespersen), the person

really running the LIS project (Paul Rabin from the OSF), and the chair of the POSIX.1 Working Group (Donn Terry) – have realised that this is POSITIVELY THE LAST CHANCE to change the ordering of the document. [Ed. — *the close() and open() functions are in different chapters of the standard, as an example.*]

Donn has come up with a potential re-ordering and this will be applied to the new documents. I was concerned that this would make balloting difficult, because we lose the ability to easily cross reference. The idea is to print a re-ordered version of 9945-1 (without rationale) to act as a balloter's aid.

The two new documents will also contain "other editorial changes." The adoption of the LIS has meant that the original text has been inspected very closely indeed, and has been found wanting in many places. It's often ambiguous with unclear wording. The text has been tightened up in these places. One of the tasks of the working group this week has been to examine a list of lines containing "may," "can," "cannot," "system defined," and some other words to ensure that they are all used consistently throughout the documents. Where ambiguities exist the wording has been repaired.

Now, you may argue that this will change the sense of the document, and it might. It will be up to the balloting group to worry about that. There are NO conscious changes.

New functionality and real bug fixes have been held over in POSIX.1a. There was no discussion on this during the week, because the person driving that, Roy McKean from X/Open, was unable to be in Dallas.

Report on POSIX.2: Shell and Utilities

David Rowley <david@mks.com> reports on the April 6-10 meeting in Dallas, TX:

Summary

Well, it looks like it's all over but the final formalities. New drafts of POSIX.2 and POSIX.2a incorporating minor editorial changes have been approved at the New Zealand meeting of ISO WG15 as Draft International Standards. They are Draft 12 of POSIX.2 and Draft 8.05 of POSIX.2a. Both POSIX.2 and POSIX.2a should go before the Standards Board in September for approval as full-use IEEE standards.

NIST is currently working on a new FIPS (Federal Information Processing Standard) for POSIX.2,

expected in draft form by early Fall 1992.

POSIX.2b work progresses, incorporating symbolic link support within a number of utilities, and a new PAX archive format.

Test assertion work continues, with the POSIX.2 work adapting to an underwhelming mock ballot. POSIX.2a test assertion work is well under way, and appears to be easier than previously thought.

Background

A brief POSIX.2 project description:

POSIX.2 is the base standard dealing with the basic shell programming language and a set of utilities required for the portability of shell scripts. It excludes most features that might be considered interactive. POSIX.2 also standardizes command-line and function interfaces related to certain POSIX.2 utilities (e.g., `popen()`, regular expressions, etc.). This part of POSIX.2, which was developed first, is sometimes known as "Dot 2 Classic."

POSIX.2a, the User Portability Extension or UPE, is a supplement to the base standard. It standardizes commands, such as `vi`, that might not appear in shell scripts, but are important enough that users must learn them on any real system. It is essentially an interactive standard, and will eventually be an optional chapter to a future draft of the base document. This approach allows the adoption of the UPE to trail Dot 2 Classic without delaying it.

Some utilities have both interactive and non-interactive features. In such cases, the UPE defines extensions from the base POSIX.2 utility. Features used both interactively and in scripts tend to be defined in the base standard.

POSIX.2b is a newly approved project which will cover extensions and new requests from other groups, such as a new file format for PAX and extensions for symbolic links.

Together, Dot 2 Classic and the UPE will make up the International Standards Organization's ISO 9945-2 – the second volume of the proposed ISO three-volume POSIX standard.

POSIX.2 Status

Draft 12 of POSIX.2 has been prepared, a minor revision of Draft 11.3 to take care of some editorial concerns for ISO WG15. This new draft will form the final POSIX.2 standard, expected to be approved at the September meeting of the IEEE

Standards Board. Draft 12 has also been approved by ISO WG15 as a Draft International Standard. It is certainly a help to implementors to have both the IEEE and ISO versions of the Shell and Utility standard coordinated in this manner.

POSIX.2a Status

In a similar fashion to POSIX.2 Classic, a minor revision of POSIX.2a has been prepared to address some minor ISO editorial concerns. Draft 8.05 (so named to reflect the extent of the changes) will form the final POSIX.2a standard, and should also be approved at the September meeting of the IEEE Standards Board. This draft has also been approved by ISO as a Draft International Standard.

FIPS and Certification

Now that NIST is preparing a new FIPS for POSIX.2 and POSIX.2a, the issue of conformance testing and certification is rearing its contentious head once again. The problem is one of timing and organization. NIST of course wishes the certification suite to be based on the POSIX.3.2 test methods work. However, it has only just gone to mock ballot, and is still quite a distance from completion. The POSIX.2a test methods work has only recently started. In spite of this, NIST wishes to put forth a FIPS now in order to encourage the use of the standard within the US Government. Unfortunately no standard metric for gauging conformance will exist for some time. NIST's lack of money for test suite efforts is causing a number of vendors concern and frustration, causing other solutions to be investigated. If you would like up to date information on the current status of POSIX.2 conformance testing, please feel free to drop me a note.

PAX File Format

The new file format for PAX is progressing, but the group is still not completely convinced that the ISO 1001 tape format is the best technology to base the format upon. No alternatives have been put forth, so the group will likely continue along the current path until someone makes a counter-proposal.

One issue decided at the Dallas meeting was the codeset to be used within the archive to represent filenames. The 16-bit plane of Unicode/ISO 10646 (UCS2) has been selected as a good reference set of glyphs which should suit the needs of the vast majority of users. A step up to UCS4, the 32-bit version, will be planned for in the format. Gary Miller (IBM), POSIX internationalization and codeset guru, has given his blessing to the approach.

Test Methods

The POSIX.3.2 Test Methods for POSIX.2 mock ballot did not go well. Hardly any comments were received, so the group spent the Dallas meeting in small groups, one group working on creating ballot objections, and another on ballot resolutions. This isn't how it's supposed to work, folks. It is critical that the test methods work has the same level of broad-based input that the POSIX.2 standard enjoyed. Although the skill set required to effectively ballot the document is specialized and rare, the effort needs as much input as possible.

The document will go out of mock ballot for a while until a plan to get reasonable feedback has been formulated.

Work on the POSIX.2a test methods also progressed. The earlier fears of the difficulty of creating assertions for the interactive commands (*vi*, *talk*, etc.) have proven to be largely unfounded. However, turning the assertions into a test suite may still be a challenge.

Report on POSIX.3: POSIX Test Methods and Conformance

Andrew Twigger <att@root.co.uk> reports on the April 6-10, 1992 meeting in Dallas, TX:

SCCT Matters

Once again, the Steering Committee for Conformance Testing (SCCT) met for three sessions during the week. During the first session, Roger Martin (Chair) announced that three new members had been invited to join the SCCT. These are Jerry Powell (IBM), Stephe Walli (MKS) and Dan Hegerty (US Navy). The four remaining members of the SCCT (Roger Martin, Lowell Johnson, Andrew Twigger, and Bruce Weiner) were appointed for a further two year period.

The SCCT realised that unless it became more pro-active in encouraging the POSIX working groups to meet their test method development plans, the groups would not complete this work item. There had been a marked drop in the requests to the SCCT for test method consultancy from the working groups. It was believed that, in several cases, test method development was being sidelined while other issues were advanced. The SCCT decided that it needed to monitor progress more regularly and to advise the Project Management Committee in cases where slippage became evident.

The SCCT also became involved in discussions about the production of test methods for language independent specifications (LIS). [Ed -

Don't groan people. This stuff has real value.] This was discussed in the context of the POSIX.1 LIS. The thinking goes as follows:

Language Independent Specifications are useful. They provide the functional specification upon which programming language syntax is layered in the language's most natural form. The intent is to allow different languages to bind as easily as possible.

Real implementations support the functionality described in the language independent functional specification through language bindings. Real implementations are only valid through real languages, and can only be tested using real languages.

In the same way that the functionality behind each language binding is the same, the test assertions are for the most part functional test assertions. There are additional syntax related assertions for each language, but a large percentage are functional assertions.

By expressing the assertions as functional assertions written to the LIS standard, real test cases in different languages can be written. *[Ed — think about the problem of verifying a POSIX.5 (Ada) run-time implementation.]*

The initial target for LI test assertions was the revised POSIX.1 LIS, which is expected to enter ballot in the next quarter. The SCCT decided that they would accept POSIX.1LIS entering ballot with a reference to the current POSIX.3.1 C language binding assertion set, but that LI test assertions would be needed before ballot could complete.

At the moment there seems to be little interest in producing the LI test assertions — the task was described as a further layer of boredom on top of an already boring task! However, the SCCT believe that there is considerable value to those working groups who are amending POSIX.1 to develop a set of LI test assertions and this really needs a base set of assertions from POSIX.1 LIS.

Test Methods for POSIX.1

POSIX.3.1 is the test methods document for POSIX.1, the base operating system service interface. During the meeting the technical reviewers worked to resolve the remaining objections against draft 13 of this standard. It is believed that all of the outstanding objections have now been dealt with, and that the document is ready for a final recirculation ballot. It is hoped that this will

be completed by the end of June and the document forwarded to the standards board shortly afterwards.

Test Methods for POSIX.2

The POSIX.3 and POSIX.2 working groups met jointly for most of the week with the available members from each of the groups starting to review the current draft document. This exercise caused many of the members of the group to realise how many areas still needed to be addressed, and at the end of the week a plan was put together to provide enough input to the technical editor to allow a much more complete draft to be produced.

Concurrent with this task, a few members of the POSIX.2 working group continued with the specification of test methods for POSIX.2a (UPE). Most of the work on the simpler utilities was completed, but the larger utilities still need to be tackled.

Report on POSIX.5: Ada Binding to POSIX.1

*Del Swanson <dswanson@email.sp.unisys.com>
reports on the April 8-12, 1992 meeting in Dallas, TX:*

The POSIX.5 group has been working to produce Ada language bindings to POSIX standards. So far, we have been concentrating on the POSIX.1 standard and the Real-time Extensions standards being developed by POSIX.4. There are informal plans to prepare a project request (PAR) to develop an Ada binding to POSIX.2 as well.

The big excitement at the Dallas meeting was that Draft 8 had been produced in a short time, fixing minor problems in Draft 7, and was sent out for a fast recirculation. This draft was overwhelmingly approved, and Draft 9, encompassing a few editorial changes, is being submitted to the Standard Review Board for its final approval as an IEEE standard. *[Ed. — Del informs me that POSIX.5 has been approved as an IEEE Standard by the Standards Board on June 18. Congratulations to all who worked on and balloted the document!]*

Meanwhile, the group proceeded blithely along with its new task, to develop an Ada binding to the Real-time extensions being balloted from POSIX.4. Three position papers had been prepared, and were presented to the group, on the relationship of Ada runtime library functionality and the Real-time extensions. The issues were outlined in the report of the last meeting.

The group was fortunate to be presented with a draft thin binding to POSIX.4, which had been prepared at Florida State University under con-

tract to the U.S. Army. The group divided up the document, and individuals presented analyses to the group. The task for the POSIX.4 Ada binding group appears to be a cooperative effort with FSU, which should speed the process significantly.

Everyone agreed that the binding to POSIX.4 will be relatively straightforward. The POSIX.4a (Threads) binding, however, will have more significant problems.

Currently we are proceeding with the Ada bindings to the Real-time extensions in the same manner used for the POSIX.1 binding, i.e., working from the C language interface. By TCOS fiat, the binding will ultimately be to the Language Independent Specification of the POSIX.4 documents. The hitch is that the Real-time extensions group is just recently moving beyond its initial experiments with LIS, done in early 1990. The pieces are all finally in place, with an LIS of POSIX.1 and its C-binding (POSIX.16), to start the work seriously. At the April session, there was significant interaction between the two groups, to try to make the transition smoother.

Two issues in particular were addressed. First, the POSIX.5 working group composed a list of elements of the C binding which we thought particularly needed to be made language neutral, and discussed them with the Real-time group. Second, since it was agreed that ideally the names of the LIS should be reflected in all the language bindings, we provided to POSIX.4 a list of identifiers which seemed appropriate for the functions.

We have also supplied to POSIX.4 a draft of the thin Ada binding to what we have projected as an LIS. The hope is that seeing the results of a binding to an LIS will provide some guidance for the development of one.

We are expecting that within a couple of more drafts the current thin binding to POSIX.4 will be in good condition. We are meanwhile dividing up the responsibilities to start on sections of POSIX.4a and POSIX.4b. It is still a bit early to project a realistic date for beginning balloting.

Report on POSIX.17- Directory Services API

Mark Hazzard <markh@rsvl.unisys.com> reports on the April 6 - 10, 1992 meeting in Dallas, TX:

Summary

Draft 3.0 of POSIX.17 began IEEE ballot on April 7th and finished the first round of balloting May 19th with 84% of the ballot group responding. We completed sending responses to all who participated in the Mock Ballot of Draft 2.0.

The group formed a ballot resolution team, and dealt with the "Which track to ISO?" issue. Splitting/re-casting our Project Authorization Request (PAR) was a hot topic. We're following a PMC recommendation to separate the Directory Services API work (which is in ballot) from the POSIX name space issue which hasn't received much attention.

Introduction

The POSIX.17 group has defined and is balloting a user to directory API (e.g., API to an X.500 DUA - Directory User Agent). We used APIA — X/Open's XDS specification as a basis for work. XDS is an object oriented interface and requires a companion specification (XOM) for object management.

XOM is a stand-alone specification with general applicability beyond the API to directory services. It is used by IEEE P1224.1 (X.400 API) and is being standardized by the P1224 working group.

The current POSIX.17 PAR has a two part scope. The first authorizes the group to work on an API to directory services. The second (and more contentious) part addresses the POSIX name space issue. The working group has discussed name space, but decided to focus on the API to directory.

POSIX.17 is one of five "networking" groups under TCOS, and comes under the purview of the Distributed Services Steering Committee (DSSC).

Status

The group finally completed all the written responses to the comments received from the Mock Ballot of Draft 2.0 of our document. If you responded, you should have a reply by now.

Draft 3.0 of POSIX.17 was distributed for IEEE ballot just prior to the Dallas meeting, and included all test methods and the language independent specification (LIS). The document grew from 234 pages in Draft 2.3 to 478 pgs in Draft 3.0 with the inclusion of all remaining test methods.

As of this writing, the 1st ballot is now officially closed, with 84% of the Ballot Group returning ballots.

Once again, we met with POSIX.12 (Protocol Independent Interfaces) in joint session and discussed their requirements on directory services. The white paper produced by POSIX.17 was used as a basis for moving ahead on requirements. (The white paper was the result of an action taken

in Irvine to document agreements, assumptions, issues, options and proposed actions.)

The meeting was quite productive and resulted in an understanding on how to progress the work. POSIX.17 took an action to assist the POSIX.12 group with writing an annex mapping a simplified, more focused interface to the POSIX.17 API.

Some POSIX.17 members met with P1224 to process the comments/objections raised during the initial round of balloting of the object management specification.

The PMC recommended in January that the POSIX.17 project request (PAR) be split into two separate projects, one for the Directory Services API work (which is in ballot) and the other for the POSIX name space issue which hasn't received much attention.

Name space conjures up many different things for different audiences. Some folks see the issue as a language issue, dealing with function prefixes and the like. The working group sees the issue as one in which objects are uniquely named in a global context, i.e. beyond a single kernel. If we use the process id as an example, we find that the 5-digit positive integer used as the name for a process within most kernels doesn't scale too well globally. If I want to have a utility that determines the status of all my processes, even those on other kernels, I have to somehow extend the name space.

There was a spirited debate as to whether or not a second PAR was needed for name space work, in that the issues could be resolved by some other mechanism in the TCOS realm. Neither POSIX.17 nor the System Interface Coordination Committee (SICC) believe that POSIX.17 owns the "C" name space issue. A white paper will be produced summarizing the name space issue and the work to date. Stay tuned ...

The road to ISO

The group spent much time debating how to progress the POSIX.17 API work for ISO standardization. The central point of contention was a proposal to remove the POSIX.17 API from ISO 9945-1 to join P1224/P1224.1 in a to-be-determined track in ISO. [Ed. — *ISO/IEC 9945-1 is the ISO name for IEEE 1003.1, or POSIX.1. All other system interfaces, such as POSIX.4 real-time and POSIX.6 security, are supposed to be integrated to 9945-1 in future amendments.*]

The rationale given was that since the POSIX.17 work is dependent on P1224 and all three documents share the same style of interface and roots,

they should all be progressed to ISO within the same Working Group. Since P1224 and P1224.1 aren't part of (and won't be part of) 9945-1, POSIX.17 should be pulled out of 9945-1 and progressed with the other two documents.

There is a risk that ISO SC22/WG15 (the ISO POSIX Subcommittee 22 Working Group) will not accept a work item for an API to directory services outside of 9945-1. The implication is that a new SC22 working group (or one from SC21 or SC18) may be required for this work, with all the associated start-up overhead. All this could delay the work and subsequently jeopardize its completion as an ISO standard.

Taking the work from 9945-1 also breaks the link requiring a distributed POSIX system to include an API to directory services. At least one other distributed services working group (POSIX.12) was concerned about this as well.

Arguments against the non-9945-1 track to ISO resulted in a compromise that will (hopefully) allow us to retain the reference to the POSIX.17 work in a work item for 9945-1. The work item could revert to a pointer to the work being done outside of 9945-1 (if that comes about) and also serve as a place holder for our work within SC22 WG15 if another track couldn't be found.

A resolution was prepared for the SEC, proposing that the SEC authorize POSIX.17 to take several actions relating to the mechanics of progressing our document through the IEEE ballot process and on to ISO. After some initial tough sledding late Thursday night, (my Minnesota roots showing), the SEC accepted all the time critical aspects of the resolution, deferring the rest until Chicago.

In Closing ...

Once again, there are quite a few homework assignments between meetings. The ballot resolution process begins. Look for a white paper rationalizing the directory services API work with the name space issue. We also need to submit a New Project proposal for progressing the POSIX.17 to ISO within SC22.

The group will meet next time in Chicago, concentrating on Ballot resolution and name space issues. We plan to meet in Utrecht and possibly for a few days in Reading, UK, to complete the work for our first (and hopefully final) ballot recirculation.

Report on P1224: X.400 API

Steve Trus <trus@duke.ncsl.nist.gov> reports on the April 8-12, 1992 meeting in Dallas, TX:

Summary

P1224 is the Object Management API, based on X/Open's Object Management specification (XOM). It is used by POSIX.17 (Directory Services API) and the P1224.1 document. P1224.1 is the X.400 API.

P1224 spent a productive meeting in Dallas, and we are very near the completion of the standardization of the P1224 and P1224.1 documents.

At the Dallas meeting we:

1. discussed our goals for the International Standardization of the IEEE Networking APIs,
2. planned future work for the P1224 group,
3. presented the status of the IEEE balloting of P1224,
4. presented the status of the IEEE balloting of P1224.1,
5. planned the recirculation of the P1224 document, and
6. resolved ballot objections and reviewed ballot comments for the P1224 document.

International Standardization of the networking APIs

We discussed options for the International Standardization of the networking APIs. The goals of the P1224 group are to have our work standardized with minimal changes in JTC1, and to have the X.400 and the POSIX.17 Directory Services APIs standardized in the same JTC1 Subcommittee.

P1224 Working Group Future Plans

Plans for standardizing future X.400 related APIs were discussed. The X.400 API Association and X/Open will have stable base documents for a P7 and an EDI API by the end of 1992. Tentatively, we would like to begin converting these documents into IEEE standards at the January 1993 meeting.

P1224 Status

Balloting of the P1224 document began January 1, 1992, and ended January 31. The ballot group consists of 73 members. The P1224 ballot closed with 87% of the ballots returned, and 75% of the eligible voters approved the document. The test methods for P1224 will be included in the first

recirculation of the document. (Balloting cannot complete until the test methods are balloted.)

The group spent two days resolving ballot objections and reviewing ballot comments for the P1224 document. The technical editor will incorporate the changes and the test methods into the document.

We agreed to limit the recirculation objections and comments to changes to draft 4 of the P1224 document and test methods. Recirculation begins May 17, 1992 and it will end June 19.

P1224.1 Status

The P1224.1 balloting period will begin May 6, 1992 and will end June 5. There are 49 people in this balloting group. The test methods will be included in the initial ballot of the P1224.1 document.

Iain Devine, the P1224 technical editor will be the ballot resolution reviewer, assisted in technical matters by members of the X.400 API Association and X/Open.

In Closing ...

The progress of the P1224 working group is very good. We hope to have the P1224 and P1224.1 standards complete by the end of 1992. The primary function of the July and October meetings will be P1224 and P1224.1 ballot resolution.

Report on The IEEE Standards Board

An Anonymous Friend of USENIX reports on the March 1992 meeting.

[Ed — Anyone wishing to send comments to the report writer may do so through me.]

The March 92 meeting of the IEEE Standards Board contained some very interesting action on the GUI project authorization requests (PARs), more forward (or backward) movement on other TCOS (POSIX) PARs, and broad developments in the IT (Information Technology) field in general.

X3/JTC1 U.S. TAG merger

One of the big discussion items on last year's Board agendas was the proposed merger of X3 with the ISO/IEC JTC1 U.S. TAG (U.S. Technical Advisory Group for ISO/IEC Joint Technical Committee 1). Considered to be an administrative advantage for both organizations, and a means to speed up the possible internationalization of U.S. IT standards, there was concern within the IEEE as to how its standards groups would be represented in the international arena.

At the March 92 meeting, it was reported that this merger in its current form did not achieve approval through a consensus vote. It is expected that work will be done on the proposed merger and that it will reappear in a future letter ballot of the JTC1 U.S. TAG (the IEEE is a member of this TAG).

IT Standards Funding

Another motion that the Board discussed is a proposal from ANSI to charge "participants" from the U.S. in international standardization efforts a fee to cover the administrative costs of handling the international IT standards activities. Remember, ANSI is the member body representing the U.S. in ISO. (For the IEC, its a group called the U.S. National Committee, or U.S.N.C). The Board created an ad-hoc committee to address this. This committee held its first meeting during Board week and explored guidelines and processes to come up with a response to this request. Gary Robinson and John Rankine are the Computer Society representatives on this committee.

Cray Users Group

Cray Users Group requested Organizational Representative status at this Board meeting and, with the recommendation of the TCOS chair, was approved as an OR by the Board.

TCOS Inside Track on RevCom

One of the TCOS vice-chairs, Lorraine Kevra, is now on the IEEE Standards Board Review Committee (RevCom), which gives recommendations for final approval of standards to the Board. Lorraine will be able to bring first-hand experience with this back to TCOS and, hopefully, be able to explain the convoluted existence of POSIX to RevCom!!

The next IEEE Standards Board meeting will be June 16-18 in San Juan, Puerto Rico. The following meeting will be September 15-17 in New York City. The deadline for submission of PARs and standards for the September meeting is August 7.

NesCom (New Standards Committee Activity)

NesCom set a new record for work, with over 75 PARs on their agenda. The meeting went for over six hours. If you could ever imagine a completely exhausted committee, NesCom was it at the end of their day!

Approved New TCOS Projects

P1003.7.1 (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 3: System Administration--Amendment: Print Administration

P1003.7.2 (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 3: System Administration--Amendment: Software Administration

P1003.16a (OS) Standard for Information Technology--POSIX C Language Interfaces --Part 1: Binding for System Application Program Interface (API)-- Amendment 1: System API Extensions

Approved TCOS PARs to Revise Existing Standards:

P1003.2b (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 2: Shell and Utilities

Approved TCOS Revised PARs:

P1003.1 (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 1: System Application Program Interface (API) [Language Independent]

P1003.1a (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 1: System Application Program Interface (API) [Language Independent]-- Amendment 1: System API Extensions

P1003.7 (OS) Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 3: System Administration Interface

P1003.16 (OS) Standard for Information Technology--POSIX C Language Interfaces--Part 1: Binding for System Application Program Interface (API)

P1201.1 (OS) Standard for Information Technology--Uniform Application Program Interface--Graphical User Interfaces

TCOS PARs for Which Approval Was Withheld

There was one unapproved TCOS PAR:

P1003.19 (OS) Standard for Information Technology--POSIX Fortran 90 Language Interfaces--Part 1: Binding for System Application Program Interface (API)

This project was not approved because the scope did not clearly imply that this standard would not change the existing language standard produced in X3. The amended PAR was not filed in time for the June Board meeting; let's hope for September!

PARs Removed From the NesCom Agenda:

P1295.1 (SCC) Standard for Information Technology--X Window System--Modular Toolkit

P1295.2 (SCC) Standard for Information Technol -

ogy--X Window System--Open Toolkit Environment

These PARs (the GUI PARs) were removed from the NesCom agenda per NesCom member John Horch because the Sponsor-approved wording changes were not available in time for the NesCom meeting. They will be reintroduced at the June Board meeting.

[Ed. — The Standards Advisory Board has apparently withdrawn the offer of hosting the sponsorship of the GUI PARs from TCOS. The supporters of the Open Toolkit Environment and Modular Toolkit PARs (Motif and Open Look by different names), have convinced the SAB their destiny lies elsewhere.

This is despite the fact that they fall within TCOS's scope statement, and that the P1201 windowing PARs lie within TCOS.]

Report on ANSI X3J11 and ISO/IEC SC22/WG14: C Language

Michael Meissner <meissner@osf.org> reports on the May 10-15, 1992 meeting in Salt Lake City, UT:

On May 10-12 of 1992, I attended the ANSI X3J11.1 meeting, and on May 13-15 of 1992, I attended the combined ANSI X3J11 and ISO WG14 meetings.

For those people who aren't aware of how the various committees interact, and what their charter is, here is a thumbnail sketch. In the beginning was the ANSI X3J11 committee, which is the American committee chartered to produce a C standard. The first C standard was approved in December of 1989, and is available as X3.159-1989. The X3J11 committee is now doing interpretations, where they have to answer queries about the standard, but cannot change it.

Around 1988, the ISO WG14 committee was formed to lead the American C standard through as an international standard. In ISO, each country gets one vote, and the USA votes through ANSI. After reformatting the standard and moving some sections around to meet ISO guidelines, the C standard was approved as an international standard, which is available as ISO/IEC:9899-1989(E).

At the time the standard was approved, there were three open issues raised by Japan, Denmark, and England, and there was approval to work on a normative addenda to address the problems. (These issues are covered later.)

Around 1989, some people started meeting to discuss numerical issues and the C language. The committee, originally called NCEG (Numerical C

Extension Group), has since become X3J11.1, a subcommittee of X3J11. Their charter is to produce a technical report, which does not have the weight of a ANSI or ISO standard. I suspect many of the X3J11.1 features will be items to be considered for the next ANSI/ISO C standard. This committee is made up of various interested parties who care about floating point calculations.

X3J11.1

X3J11.1 met for the first three days, from May 10 through May 12.

I went to the floating point extensions subgroup on Sunday night. For the most part, this meeting was uncontroversial. The floating point extensions group had submitted their draft to a letter ballot which passed, and the meeting was used to address minor editorial changes and comments from the ballot. The draft contains the following items:

New syntax for floating point constants, so that you can specify the exponent and mantissa in hexadecimal, rather than decimal.

Printf/scanf %a/%A format specifiers to print the floating number in the new hexadecimal format.

More math functions.

Overloaded math functions — these functions are a step towards C++ style overloading: if the arguments are single precision, the calculation is done in single precision. Unlike C++, these are only required for the system functions and not the user functions.

Requirements on exactly when Nan/Infinity/-0 is produced from the various match functions if the system uses IEEE 754/854 floating point. (Most systems these days use IEEE 754 format).

Adding IEEE unordered comparisons (!>, etc.) which return true if either value is a Nan, instead of false.

Adding floating point classification functions.

Ways to get/set exception flags.

Two new include files are added.

On Monday and Tuesday, I went to the normal X3J11.1 meetings. The following items were discussed:

The restricted type qualifier proposal had a successful letter ballot, and will go outside of X3J11.1 for review. This proposal is halfway between the current situation where the compiler can't fully vectorize, and noalias, which got shot down before the standard went out. It adds a new qualifier, restricted, which says that you promise that the given pointer is the only way a particular item is referenced. This will allow a function to take two restricted pointers, and to fully vectorize the accesses, because the compiler doesn't have to worry about overlap cases.

Automatic variables with variable dimensions were discussed, but no conclusion was reached. There are two proposals on the floor, one from Cray and the other from USL. The Cray proposal would require people to pass the bounds explicitly for arrays, and has problems in scoping if the bound is passed after the array. The USL proposal which is authored by Dennis Ritchie, would pass a "fat" pointer, which is a descriptor that contains the bounds as well as the pointer. The debate went on as to which was more in the spirit of C. I personally tend to favor the USL proposal.

Designated initializers will go out for a review. These allow a programmer to initialize a structure or array out of order. For example:

```
struct foo {
    int a, b;
} st = {
    .b = 1, .a = 2
};
int foo2[10] = { 1, [5] = 2, 3 };
```

(In the array example, element 6 is initialized to '3'). Gcc 2.0 has a similar feature, though the syntax is slightly different.

Compound literals will go out for a review. These allow a programmer to create an automatic (or static if at file scope) aggregate without having to give it a name. Gcc has this feature. For example:

```
foo (&(struct bar){ 1, 2 });
```

The floating point extensions draft mentioned earlier was approved to go out for a review. One item that will go in a cover letter is to warn people that the #pragmas specified may be changed into macros, since pragmas are not allowed inside macro expansions.

The complex arithmetic draft was not ready to be sent out for review at this time. The draft needs to be more fully specified for IEEE floating point with respect to Nans and Infinities. Also, there was concern that the complex functions be folded in with the overloaded functions (ie, having just

sin instead of csin). Finally, some people feel that in addition to real, and complex types, there needs to be an imaginary type that has no real component, particularly in the case with Nans and Infinities.

There was some spirited discussion about extended integers and 64 bit machines. The 64-bit consortium (vendors who will be producing 64 bit CPUs) want the ANSI group to exactly specify what sizes short, int, long, etc. are in 64 bit environments. Given that ANSI committees typically take years to come down from the mountain, and the 64-bit consortium needs to deliver products soon, it was hopeless. Also, there are good reasons why the standard only gives minimums. The crux of the problem is that when you move to 64 bits, programs will break (just like they did in moving 16 bits to 32 bits, but there is more extant code in C now). No matter what you choose, you break somebody's cherished notations. One camp wants int, pointer size, and long to all be 64 bits, and there is no explicit 32 bit type. Another camp wants int to be 32 bits, and pointers/long to be 64 bits. Finally at least one person wanted int to be 64 bits and long to be 32 bits. The C committee roundly reviled any rule that broke the rule that sizeof(int) <= sizeof(long), but otherwise had no comments to send back to the 64-bit consortium. The array syntax subgroup met on Monday night. This group is charged with doing things to arrays, so that fast code can be generated on the vectorizers and/or massively parallel machines (essentially Cray vs. Thinking Machines).

The meeting quickly broke down into shouting matches and such. I felt that it made negative progress, to the point that the only positive vote was a "motherhood" vote on the group's charter. There was another array syntax subcommittee meeting on Tuesday night (and possibly Wednesday night also), but I declined to attend.

NSI X3J11/ISO WG14

On Wednesday through Friday (May 13 - 15), the ANSI X3J11 and ISO WG14 met together. At times the meeting was run in ANSI X3J11 mode, and at other times it was in ISO WG14 mode. The primary objective for the ANSI part of the meeting was to answer questions about the standard. The primary objective of the ISO part of the meeting was to deal with the three proposed normative addendum.

The U.K. addendum is designed to tighten up the wording of the standard, but not to make any substantive changes. The goal of the Japanese addendum is to add additional wide character

functions and a new header in which to declare them. The Danish addendum provides alternatives to the ANSI trigraphs, while not using any of the national replacement characters from ISO 646.

The big news is that the ANSI C standard will soon be withdrawn and replaced with the ISO C standard, so that the standards remain synchronized. This means that chapter and verse quotations will soon change, due to paragraph renumbering required by ISO. Also, when the normative addenda come out, they will become part of the ANSI C standard, in addition to the ISO C standard.

Some of the decisions reached in talking about the Japanese addenda include:

Wide character I/O functions can return errors if they can't translate multibyte <-> wide characters. Errno is set to CEILSEQ upon such an error.

If a wide character value is ≥ 0 and \leq UCHAR_MAX, then the single byte character classification functions (*isprint()*, *isspace()*, etc.) if true, implies that the wide version (*iswprint()*, *iswspace()*, etc.) is also true. If the single byte version is false, it does not imply that the wide version also returns false. This is to allow wide characters to fill up positions in the encoding that aren't valid single byte values.

We voted against adding more support for mixing multibyte and wide character strings in the **printf()*/**scanf()* family of functions. The proposal was for %hs to always mean multibyte characters in both *printf()* and *wprintf()*, %ls would always mean wide

characters, and %s would mean either multibyte or wide characters, depending on whether the function was *printf()* or *wprintf()*.

The new function *wcswcs()* (wide version of *strchr()*), got renamed to *wcsstr()*, since most people felt that the second 'str' represented substring.

We voted not to reserve the wide stdio functions for a future standard to put in *stdio.h* (ie, you always have to include *wchar.h* to properly declare those functions).

We voted that no illegal multibyte sequence will be emitted by the wide character output routines (including through %S or %C in *printf()*).

We voted that only a single byte space terminates *scanf(' '%S'')*, ie. not *iswspace()*, to allow for logically ungeting just a single byte.

The Danish digraph proposal was shot down (again). I suspect it may be for the last time, because more countries are concerned about delaying the rest of the addenda for this one small issue. Japan and the Netherlands both voiced this opinion for the first time at this meeting.

There will be letter ballots sent out on the various responses to interpretation requests. One letter ballot will cover all decisions in which there were no "no" votes at the committee, and one letter ballot will be sent out for each decision that had at least one "no" vote. It is hoped that the draft for the document of interpretation requests will be passed in the letter ballot, so it can be sent out for the next meeting (6 months from now).

ISO Monitor Report

ISO Monitor Report on the May 1992 ISO POSIX Meeting

by Stephen Walli
<stephe@mks.com>

Overview

The International Standards Organisation (ISO) and the International Electrotechnical Commission (IEC) jointly develop international standards for information technology. The family of IEEE standards known as POSIX are being brought forward as international standards.

The ISO view of this process is that the standards are being developed by a national body (U.S.) instead of the more traditional model of ISO working group development. (Similar national body development is going on for C++ in JTC1/SC22/WG21 which meets jointly with ANSI sponsored X3J16.) The IEEE forwards work through an ANSI sponsored Technical Advisory Group (TAG), to ISO/IEC JTC1/SC22/WG15. This frightfully long agglomeration of acronyms stands for ISO/IEC Joint Technical Committee 1 (JTC1), Subcommittee 22 (SC22) on Programming Languages, Working Group 15 (WG15) on POSIX.

WG15 (as we shall refer to it) helps guide the IEEE documents as they come forward as ISO standards. Direct development of the documents does not happen in WG15, but rather it acts as a focal point for international comment and much of the liaison work that is required to ensure that the IEEE documents will be able to stand as ISO standards.

The point of the process is to develop a single standard which does not diverge from the IEEE counterpart. The groups have succeeded to date, with the base operating system API embodied by IEEE Std 1003.1-1990 being identical to ISO/IEC 9945-1:1990 with the minor exception of the plain white ISO book cover. The IEEE Standards Press even produces the ISO book, and they do so on A4 paper no less!

The WG15 projects are organised into three standards: 9945-1 represents all of the operating system APIs, 9945-2 represents the shell and utilities, and 9945-3 will be the system administration functionality.

Currently, the IEEE POSIX.4 (Real-time), POSIX.6 (Security), and POSIX.8 (Transparent File Access) documents are all somewhere in the WG15 review-and-comment process. These documents will all be rolled (as programming language independent functional specifications) into 9945-1. POSIX.2 and POSIX.2a will become 9945-2 in the (relatively) near future. POSIX.7.1 (Printer Administration) is making its debut on the ISO WG15 scene this meeting in a very informal way, as the WG15 members were encouraged to join the initial mock ballot. This book will eventually become part of 9945-3.

The last thing worth mentioning before getting into the report of this meeting is the group itself. There were 21 attendees. (The IEEE typically has around 350 attendees.) This number is a little low, as we were meeting on the other side of the globe in New Zealand. These 21 people represented 9 countries (one country gets one vote.) Size of delegation is always fun to note. (Please see the table.)

Country	Count	IEEE
U.S.	4	4
Canada	4	2
England	2	2
Germany	1	1
France	1	-
Italy	1	-
Japan	1	-
Denmark	1	-
New Zealand	4	-
Officers	2	2
9	21	11

The officers are the convener (Jim Isaak, U.S.) and the project technical editor (Hal Jespersen, U.S.). The overlap is also interesting. Jim Isaak is both chair of the IEEE Technical Committee on Operating Systems - Standards Subcommittee (TCOS-SS), the group responsible for building the POSIX documents, as well as ISO WG15 convenor. Hal Jespersen is also TCOS-SS Vice Chair of Technical Editing, and chair of IEEE POSIX.2 (Shell and Utilities).

The other American delegates are all voting members of the TCOS-SS Sponsor Executive Committee as well, representing the Chair of IEEE POSIX.1, the Chair of the Steering Committee for Conformance Testing, the Uniforum Institutional Representative, and Vice-Chair of Logistics. One of the English delegates is Chair of POSIX.7 (System Administration). The German

delegate is Vice Chair of POSIX.6 (Security). One of the Canadians (the author) is the EurOpen Institutional Representative.

This overlap proves useful since the size of IEEE POSIX (ap 350 members) makes it almost impossible to completely overlap the WG15 and IEEE TCOS-SS meetings, as the C++ people do. There just aren't enough hours in a day for all the coordination meetings. The best that can be currently done is to run one WG15 meeting a year right beside an IEEE meeting. WG15 meets twice a year. TCOS-SS meets four times a year.

The next WG15 meeting will be in Reading, U.K., October 27-30, 1992, following the IEEE meeting in Utrecht, NL, October 19-23.

Enough of this didactic rambling. On to the report!

The Meeting

This meeting was held in Hamilton, New Zealand, as WG15 travelled to the far side of the globe in the hopes of encouraging future participation from New Zealand. Before everyone starts the "exotic locations" routine, let me point out it is 19 hours by plane for someone from the east coast of North America, with a brief (2 hour stop) in a transit lounge. Our accommodations were undergraduate (!) dormitories at the University of Waikato, who hosted the meeting. You remember undergrad dorms, a bed, a desk, a narrow aisle between them in which to dress, and the W.C. down the hall. The cafeteria (!!) food wasn't all that bad, but....

POSIX.2

One of the primary accomplishments of the week was the acceptance of POSIX.2 (Shell and Utilities) and the POSIX.2a (User Portability Extension) as a Draft International Standard (DIS). Through the hard work of Hal Jespersen, as chair of POSIX.2 and the project technical editor of both the ISO and IEEE working groups, WG15 was able to settle on a draft of the documents which met with everyone's approval.

The POSIX.2a User Portability Extension (UPE) is an amendment of the base POSIX.2 document. The two will be rolled together now.

With a little luck and optimism, the schedule should work something like this:

Summer, 1992 — Final recirculation of the two documents in the IEEE balloting group. This will be similar to the final editorial circulation of POSIX.1a as a reformatted IEEE Std. 1003.1-1988,

just prior to becoming IEEE Std. 1003.1-1990 and ISO/IEC 9945-1:1990.

September, 1992 — the two documents come forward to the IEEE Standards Board for final approval as IEEE standards (IEEE Std. 1003.2-1992).

Fall, 1992 — The combined book (ap 1400 pages!) will be recirculated for one last ballot at the international level. This ballot changes 9945-2 from a DIS to a full International Standard (IS).

Because of its sheer size (volume?), there will still be ballot objections. There is just too much being covered to have people who are happy with all of it. There are still areas which have demonstrable problems. These can and will be fixed in future amendments. We are finally down to the wire for a document that because of the breadth of its coverage has been in ballot for four years. The community is finally going to get the companion standard to 9945-1 (POSIX.1) that it wants and needs.

LIS

One of the requirements placed on the IEEE working groups forwarding API documents as standards to ISO, was that they be forwarded as programming language independent functional specifications (LIS), with at least one language binding. The intent of this method is to allow other languages to bind to the functional specification in a manner most natural to the language, and not merely re-cast the original standard's programming language syntax into something in a new language. (No one wants to propagate the GKS API that demonstrated that one could write Fortran in any language.)

There is currently an LIS version of POSIX.1, with a C binding. This was built from the original C-based 1003.1-1990. (These documents are referred to as POSIX.1/LIS and POSIX.16.) They are about to go to IEEE ballot this Summer.

Originally, these two new documents were to be an exact mapping to 1003.1-1990. The organization of the original left a little to be desired. The open() function and the close() function are in different chapters. At the New Zealand meeting, WG15 voted to allow the POSIX.1/LIS and POSIX.16 technical editor to re-organize the work based upon a new organization agreed to by all.

Additionally, it was agreed that small bug fixes should be allowed to the documents. The timing of ballots is such that it could be a long time before another round of changes comes along to "fix" the POSIX.1 book.

A concern was raised that we are opening a nasty hole into which many things will find their way. Bug fixes and wording changes (based on interpretations) are small. New functionality is not. This is something that the balloting groups will have to watch out for. As help for the balloter, two things will be added to the balloting package.

A mini 1003.1-1990, without the rationale and annexes, and reorganised to the new sections, will be sent out to allow balloters to see how the LIS and C binding align with the C-based original.

A list of all changes for bug fixes will be sent to allow balloters to quickly locate material that has actually changed in content from the C-based original.

A request has been made by ISO SC22/WG11 (Language Bindings) to bring the IEEE TCOS-SS Guidelines document that describing how to build LIS and language bindings, forward as an ISO Technical Report. The new work item request will be brought forward in the Fall meeting.

Profiling Activities

POSIX profiling work is continuing to gain acceptance in the WG15 arena. Profiles are seen by some to be the way that all the open systems standards will be put together to form coherent working environments.

WG15 has created a Rapporteur Group for the Coordination of Profiling Activities (RGCPA) to handle activities relating to POSIX profiles within ISO. (Rapporteur groups are a essentially a formal special interest group within an ISO Working Group, which acts as an official point of coordination.) RGCPA has met twice now, once last Fall and again in January.

The terms of reference for the group were established at this meeting. The RGCPA's most important role will be as a liaison point for other profiling activities within the open systems world.

The European Workshop on Open Systems (EWOS) has done some good work in determining just how to build useful profiles. Luigi Bertuzzi, representing Italy at this WG15 meeting, has been involved in this work and presented it to WG15. The EWOS work involves a number of steps to help shape a functional profile from user requirements, applying standards only as the last step. It does not try to cram user requirements onto standards, nor make the mistake of assuming the standards represent user requirements.

The IEEE POSIX.0 (Guide to Open Systems Environments) also contains profile related work. This document is about to be balloted at the IEEE level. POSIX.0 is to be brought forward as an ISO technical report as well. This WG15 meeting was the beginning of that process.

Internationalisation (i18n)

Internationalisation (i18n) is an obvious interest to an ISO standards body. WG15 created a rapporteur group on i18n for POSIX early on in its existence. WG20 is another SC22 (Programming Languages) working group which concerns itself with i18n issues with respect to programming languages in general. Keld Simenson (DK), as a member of both groups, acts as the liaison in both directions between the groups.

[One member quietly suggested we should really be concerned with intergalacticalisation. The two of us quickly coined the term "i20n". When we make first contact, remember, you heard it here first.]

WG15 forwarded a liaison statement to WG20 (Internationalisation). One of the important points of the statement was the recognition of the fact that while internationalising an application is a good thing to do, and a common portable method of doing so is a good thing to have, internationalising an application probably reduces its portability. One can very quickly add a lot of requirements to the portability of an application by internationalising it.

The Bookworm

by Peter H. Salus
Sun User Group
<peter@sug.org>

X Window Books

I bet there are more books on X now than on C++!

There's that series of around a dozen volumes from O'Reilly Associates (ORA) – definitely intended for the programmers. ORA also publishes *The X Resource*, a quarterly made up of articles and documentation otherwise totally unavailable.

When I began using X11R4, I relied on Niall Mansfield's *The X Window System: A User's Guide*. Last month I got Steven Mikes' *X Window System Program Design and Development*. In some ways this is a superior book. It is clearly written and it contains a great deal of information that will enable the relative neophyte to do work with X. Unfortunately, though released in January of this year, Mikes' volume talks about X11R4. Furthermore, though Chapter 1 begins with "What is X?", the response ("X is the name given to what is becoming the de facto standard in windowing systems.") is not the kind of answer I had expected. While the sections on User Interface Design, Resources, Widgets, Clients, and Interclient Communication appear satisfactory, arcane matters like V, W, or X10 don't even appear in the Index. This is OK as a beginner's cook book, but don't expect true explanation.

ORA was first into the fray with its *Programmer's Supplement of X11R5* last October. We now have Digital Press' massive new Third Edition of *X Window System* by Robert W. Sheifler and James Gettys (with Jim Flowers and David Rosenthal). It is nearly a kilopage in length, but it appears to be a fairly complete reference to Xlib, X Protocol, ICCCM, and XLFD. Digital Press has also come out with Marshall Brain's *Motif Programming*, but I just don't have the time to evaluate how it stacks up against the Prentice-Hall volumes (which appear under OSF's aegis) or the ones from ORA.

Issue #2 of *The X Resource* contains, among other things, an extremely interesting article on "Font Formats and Utilities" by Miles O'Neal and Dinah McNutt. As someone interested in print, I found the material on reading, converting, and creating font file formats most valuable.

C++ Programming

Whether you agree with Tom Cargill's views on multiple inheritance or not, there's no question about his abilities where C++ programming is concerned. It's a pleasure to look at his new book, *C++ Programming Style*. Modelled on Kernighan and Plauger's work, Cargill's book contains chapters on Abstraction, Consistency, Unnecessary Inheritance, Virtual Functions, Operator Overloading, Wrappers, Efficiency, A Case Study, and Multiple Inheritance. There is also a "Summary of Rules" and an Index.

MAKE

Mastering MAKE by Clovis L. Tondo, Andrew Nathanson, and Eden Yount is a pleasant small book. It purports to give me information on NMAKE (Microsoft), MAKE (Borland), and "MAKE on UNIX systems." Too much of the volume is devoted to DOS for my taste. There are a few useful items here and there, but not enough specifically UNIX material to make the book worthwhile. My on-line man pages give me more hard information. And, unfortunately, there's nothing on mk or any of the other "makes" that are around.

Quick Reference

M.S. Vassiliou and J.A. Orenstein have produced *A Computer Professional's Quick Reference*. The first 75 pages list "Common Operating System Commands" for UNIX (pp. 3-14); VAX-VMS (15-24); MVS-TSO (pp. 25-35), VM-CMS (pp. 37-43), MS-DOS (45-60), and Macintosh (pp. 61-76). My personal feeling is that MS-DOS and Macintosh need more pages than UNIX, but actually I found most of this a waste: either one knows the commands or one gets a real introductory book. There is a nice chapter on the various ISO 8859 alphabets and a chapter on standards. What's there is OK, but don't rely on it. For example, the listing of ANSI X3 committees is incomplete (it looks as though it was derived from the August 1989 list) and the other relevant committees (like Z39) aren't there at all. Cargill's excellent *Information Technology Standardization* is absent from the bibliography. Not my type of reference at all.

Property Rights

My guess is that everyone interested in software copyright, patenting, or trademarking will want a copy of *Finding a Balance: Computer Software, Intellectual Property and the Challenge of Technological Change*, the May 1992 report from the US Congressional Office of Technology Assessment. If you've been following the various Apple or

Lotus suits, the FSF's documents, or attended one of Dan Appelman's tutorials, this badly-written 225 page volume is for you. Written in bureaucratese by a committee, I did not find *Finding the Balance* soporific at all. The policy issues involved in software copyright protection, in patent protection for algorithms, and the complications faced by libraries in a world of ever-increasing digital information are discussed in detail. If Congress does want to do something, one of the first things it might do, says OTA, is clarify the scope of copyright to either include or exclude "one or more aspects of software, such as" computer languages, algorithms, design specifications, user interfaces and other interfaces.

Congress could do this by:

- Expanding upon the Copyright Law's current language on "subject matter of copyright" by saying that the above are or are not copyrightable subject material, or
- Exempting computer programs from copyright and make them subject to new "sui generis" laws.

Computer-related patents pose a special problem to the Patent and Trademark Office (PTO), OTA says, because the Supreme Court has ruled that mathematical algorithms may not be patented but processes – including processes that involve computers – may be patented.

On the question of whether or not the PTO procedures are working now, OTA concludes that they aren't. OTA states that the biggest problem preventing the PTO from carrying out its current mission is dealing with prior art. PTO is forbidden from issuing patents unless they are "non-obvious" to practitioners in the field and "novel" – that is, have never been implemented before.

Because of PTO's problems, OTA says, patents have been issued that are neither non-obvious nor novel.

The PTO has "serious" problems, OTA says, including:

- Examiner training and turnover
- Length of pendency periods (from filing to issuance for patent applications.
- The backlog of applications
- The quality and extent of the prior art database.

One of the most interesting things in the report, is that OTA asked PTO to walk it through a typical software-related patent application. PTO refused. If that isn't an admission of just how big the PTO's problems are, I don't know what is.

This is stuff that is important to all of us. If your

congressperson is one of the few who can read, the problems OTA's report points out are worth bringing to her/his attention.

Freebies

I thought the best freebie I got at UniForum in San Francisco was *The World of Standards* from the 88open Consortium. It is made up of just over 100 pages of one-page summaries of things like ISO 8348 and MIL-M-28001 and IRDS, along with references for further information, as well as summaries of the activities of CCITT, ECMA, NIST, etc. I was told that 88open would give copies away on a first-come-first-served basis until they ran out (+1 408-436-6600).

Another worthwhile freebie, which I picked up at USENIX in San Antonio, is Brad Templeton's *Clarinet Electronic Newspaper User Manual*. In addition to listing a number of available groups – my .newsrsrc lists over 3200 right now – Templeton has written a succinct guide to news and features for those who are afraid of the information deluge. Brad has told me that they give copies to customers and that there might be a few available for prospective ones.

The X Resource (O'Reilly & Associates, ISSN 1058-5591)

The X Window System: A User's Guide, Niall Mansfield (Addison-Wesley, 1991, ISBN 0-201-56344-4)

X Window System Program Design and Development, Steven Mikes (Addison-Wesley, 1992, ISBN 0-201-55077-6, 304 pp.; \$26.95)

Programmer's Supplement of X11R5 (O'Reilly & Associates, ISBN 0-937175-86-2)

X Window System Robert W. Sheifler and James Gettys (Digital Press, Third Edition, ISBN 1-5555-088-2, 1000 pp.)

Motif Programming, Marshall Brain (Digital Press, 1992, ISBN 1-55558-089-0)

C++ Programming Style, Tom Cargill (Addison-Wesley, 1992, 225pp.)

Mastering MAKE, Clovis L. Tondo, Andrew Nathanson, and Eden Yount (Prentice-Hall, ISBN 0-13-554619-2, 143pp.)

A Computer Professional's Quick Reference, M.S. Vassiliou and J.A. Orenstein (McGraw-Hill, ISBN 0-07-067212-1, 266pp.)

Finding a Balance: Computer Software, Intellectual Property and the Challenge of Technological Change, US Government Printing Office [OTA-TCT-527, ISBN 0-16-036188-5], \$11

Zen and the Art of the Internet

Zen and the Art of Internet: A Beginner's Guide to the Internet by Brendan Kehoe

Reviewed by Billy Barron¹
<billy@vaxb.acs.unt.edu>

Zen and the Art of the Internet is a new guide to the Internet that was written by Brendan Kehoe of Widener University. The author's goal is to introduce the reader to the resources that are available on the Internet while trying to avoid system specific information. It should be noted that parts of this guide were derived from other works.

This "book" is currently available via ftp (see availability below), and will be published soon by Prentice-Hall.

Zen and the Art of the Internet, which starts off with a chapter on network basics, is a good introduction to the Internet, but not a general guide to networking. Rather, it is Internet and TCP/IP specific. If this chapter can be faulted for anything, it is that it oversimplifies some of the material. On the other hand, it definitely should not scare off the novice user.

The e-mail and FTP chapters are very good, although they do get technical at times. The e-mail chapter could be improved by the addition of a section on etiquette similar to the excellent one in the FTP chapter.

The Telnet chapter is packed with examples of Telnet-accessible services, and it explains how to find out about more services. I was rather disappointed by the omission of any information on tn3270. A description of how Telnet is different on IBM mainframes is also needed. These omissions may lead to some confusion on the part of IBM mainframe users. Kehoe also describes other tools that are available on the Internet. These descriptions are well-rounded and useful, but Kehoe has just covered the most common tools.

One of the most outstanding sections of this guide is called "Things You'll Hear About." In a lot of ways, this chapter is a FAQ (Frequently Asked Questions) to the Internet, and it will

answer many questions asked by the new network user. It introduces the novice user to the folklore of the Internet without being intimidating.

Zen and the Art of the Internet also has useful sections that contain information about commercial services, other networks, how to retrieve files, and how to find out more about the Internet. The USENET chapter does a great job of covering the most common misconceptions people have about that network. The document includes a helpful glossary.

The conclusion states "this guide is far from complete--the Internet changes on a daily (if not hourly) basis." For this guide to have lasting usefulness, it will need to be updated on a fairly regular basis. From what I can tell, it sounds like Kehoe is planning to do so. I'm sending in my suggestions, and I recommend you do the same.

Overall, I was very impressed with this document. In fact, the same day that I downloaded it I had our receptionist make copies and distribute them to the whole Academic Computing Support Staff. I am going to do the same for our library. I'm giving the new user of the Internet two sources of information to start with: the first is HYTELNET and the second is going to be *Zen and the Art of the Internet*. It has a few rough spots, but I'm sure that Kehoe will fix them. Its biggest problem is that it paints too rosy a picture of the Internet, but this kind of document is intended to get users interested in using the network, not as a critique of it.

I try to stay ahead of most Internet users in terms of my knowledge of what's available and how to access it. Well, I learned a couple of things while reading this guide, so it is not just for new users. My message to Brendan Kehoe is: Keep up the good work!

Access Instructions

The file is available on host FTP.CS.WIDENER.EDU (147.31.254.132) in the directory /pub/zen and on FTP.UU.NET in (137.39.1.9) in the directory /inet/doc. Although the author reports that he has an agreement with a publisher, he has indicated that the network versions will continue to be available.

1. This article is reprinted from The Public-Access Computer Systems Review 3, no. 1, with permission.

Third UNIX Security Symposium

Baltimore, MD September 14-16, 1992

Sponsored by USENIX in cooperation with the Computer Emergency Response Team (CERT)

The goal of this symposium is to bring together security practitioners, system administrators, system programmers, and anyone with an interest in computer security as it relates to networks and the UNIX operating system. The symposium will consist of a broad range of topics including tutorials appropriate for a technical audience, peer-reviewed technical presentations, and panel sessions. Attendees will have a unique opportunity to share their experiences and ideas on UNIX system security.

Tutorial Program Monday, September 14

Network Security: The Kerberos Approach

Dan Geer, Geer Zolot Associates and Jon A. Rochlis, MIT

Intended Audience: Systems developers responsible for networked workstation environments, particularly those whose environments may include networks which are not themselves physically secure (i.e., "open" networks) and systems managers concerned about the inherent lack of security for managing today's network-based environments (e.g., UNIX's *.rhosts* files).

The amazing and constantly growing numbers of machines and users ensures that untrustworthy individuals have full access to the Internet. Given the increasing importance of the information transmitted, it is imperative to consider the basic security issues present as large open networks replace isolated timesharing systems.

This tutorial will focus on the challenges of providing security for cooperative work arrangements consistent with the location and scale independence of today's open networking environment. Attendees will gain an understanding of the kinds of security threats which result from operating in an open environment, such as one composed of a network of workstations and supporting servers. Effective approaches to meeting these threats will be presented. Although empha-

sis will be on the Kerberos system developed at MIT, public key techniques for ensuring privacy and authentication on an open network will be explored. The X.509 authentication model and the new Internet Privacy Enhanced Electronic Mail RFCs will be discussed.

Internet System Administrator's Tutorial

Ed DeHart and Barb Fraser, Computer Emergency Response Team

Intended Audience: This tutorial is designed for users and system administrators of UNIX systems. It is especially suited for system administrators of UNIX systems connected to a wide area network based on TCP/IP such as the Internet. Some system administrator experience is assumed.

The information presented in this tutorial is based on incidents reported to the Computer Emergency Response Team. The topics covered include:

System administration – defensive strategies

- Password selection
- Default login shell for unused accounts
- Network daemon configuration
- Verification of system programs
- System configuration files
- Searching for hidden intruder files
- Staying current with software releases
- Standard accounting files
- NFS configuration

System administration – offensive strategies

- COPS
- /bin/passwd replacement programs
- TCP/IP packet filtering
- TCP/IP daemon wrapper programs
- Security in programming

Site-specific security policies

- Maintaining good security at your site
- Providing guidance to users
- Handling incidents in an effective orderly fashion
- Reviewing Site Security Policy Handbook (RFC 1244)

Incident handling

- What to do if your site is broken into?

TECHNICAL PROGRAM

Tuesday, September 15

8:30 - 8:45 Opening Remarks

8:45 - 10:15 Keynote Address by Scott Charney on
The Justice Department's Computer Crime Initiative

10:35 - 12:05 WAR STORIES

There Be Dragons, *Steve Bellovin, AT&T Bell Laboratories*

The Greatest Cracker-Case in Denmark: The Detecting, Tracing, and Arresting of Two International Crackers, *Joergen Bo Madsen, The Danish Computing Center for Research and Education*

Experiences of Internet Security in Italy
Alessandro Berni, Paolo Franchi, Joy Marino, University of Genova

1:30 - 3:00 TCP/IP NETWORK SECURITY

An Internet Gatekeeper, *Herve Schauer, Christophe Wolfhugel, Herve Schauer Consultants*

Network (In)Security Through IP Packet Filtering
D. Brent Chapman, Great Circle Associates

SOCKS, *David Koblas, Independent Consultant*
Michelle R. Kolas, Computer Sciences Corporation

3:20 - 5:20 TOOLS

TCP WRAPPER, a Tool for Network Monitoring, Access Control and for Setting up Booby Traps,
Wietse Venema, Eindhoven University of Technology

Restricting Network Access to System Daemons Under SunOS, *William LeFebvre, Northwestern University*

Centralized System Monitoring with Swatch
Stephen E. Hansen, E. Todd Atkins, Stanford University

Security Aspects of a UNIX PEM Implementation
James M. Galvin, David M. Balenson, Trusted Information Systems, Inc.

Wednesday, September 16

9:00 - 10:30 TOOLS 2

Introduction to the Shadow Password Suite,
John F. Haugh, II, Locus Computing Corporation

Giving Customers the Tools to Protect Themselves, *Shabbir J. Safdar, Purdue University*

ESSENCE: A Knowledge Based Security Monitor
Linda Baillie, Gary W. Hoglund, Lisa Jansen, Eduardo M. Valcarce, Digital Equipment Corporation

10:50 - 12:20 TOOLS 2 (cont.)

Anatomy of a Proactive Password Changer,
Matt Bishop, Dartmouth College

Audit: A Policy Driven Security Checker for a Heterogeneous Environment, *Bjorn Satdeva, /sys/admin, inc.*

Secure Superuser Access Via the Internet, *Darrell Suggs, Clemson University*

1:45 - 3:15 TRACK 1 - APPLIED RESEARCH

Specifying and Checking UNIX Security Constraints, *Allan Heydon, DEC Systems Research Center; J. D. Tygar, Carnegie Mellon University*

A Secure Public Network Access Mechanism
J. David Thompson, Science Applications International Corp. Kate Arndt, The MITRE Corporation

Network Security Via Private-Key Certificates
Don Davis, Geer Zolot Associates, Ralph Swick, Digital Equipment Corporation

1:45 - 3:15 TRACK 2 - MLS

POSIX 1003.6, *Mike Ressler, Bellcore*

Is There a C2 UNIX System in the House?
Jeremy Epstein, TRW Systems Division

Software Security for a Network Storage Service,
Rena A. Haynes, Suzanne M. Kelly, Sandia National Laboratories

3:35 - 5:35 TRACK 1 - APPLIED RESEARCH (cont.)

SunOS, C2 and Kerberos - A Comparative Review, *John N. Stewart, Syracuse University*

Heterogeneous Intra-Domain Authentication,
Bart De Decker, Els Van Herreweghen, Frank Piesens, K.U.Leuven

Observations on Reusable Password Choices,
Eugene Spafford, Purdue University

POSIX Report, *Mike Ressler, Bellcore*

3:35 - 5:35 TRACK 2 - MLS (cont.)

Reconciling a Formal Model and a Prototype Implementation: Lessons Learned in Implementing the ORGCON Policy, *Marshall Abrams, Leonard LaPadula, Manette Lazear, Ingrid Olson, The MITRE Corporation*

UNIX Operating Services on a Multilevel Secure Machine, *Bruno d'Ausbourg, CERT/ONERA France*

Distributed Trusted UNIX Systems, *Charisse Castagnoli, Charles Watt, SecureWare, Inc.*

Standards Update

Program Committee

Ed DeHart, Program Chair *CERT*

Matt Bishop
Bill Cheswick
Ana Maria De Alvare
Jim Ellis
Barbara Fraser
Ken van Wyk

*Dartmouth College
AT&T Bell Labs
Silicon Graphics
CERT
CERT
CERT*

For information on hotels and registration, please contact the USENIX Conference office.

USENIX Systems Administration Conference (LISA VI)

Long Beach, CA October 19-23, 1992

This year's LISA conference has been expanded to five days (the week of October 19-23). The conference committee has attempted to gear the program towards system administrators from sites of all sizes, both large and small.

The program will have most of the "features" of the USENIX main technical conferences: a terminal room with access to the Internet, two days of tutorials, three days of technical presentations, an "invited talks" track, Birds of a Feather sessions, a vendor display, and much more.

Preliminary Tutorial Program Monday and Tuesday, October 19 and 20

The LISA tutorial program will offer introductory as well as advanced, practical tutorials. Courses are presented by skilled teachers who are hands-on experts in their topic areas. The LISA tutorial program has been developed to meet the needs of an audience of novice through experienced computer professionals.

Attend the tutorials and benefit from this opportunity for in-depth exploration and skills development in essential areas of UNIX system administration. Combining the two-day tutorial program with the three days of technical sessions give attendees the opportunity to learn from experts at a convenient time and at a reasonable cost.

The tutorial program is divided into three tracks of half-day tutorials. Attendees may select from any non-overlapping tracks. Although some prior knowledge may be needed for the advanced

tutorials, each tutorial is presented as a stand-alone class (for example, a student may take "X and the Administrator - part 2" without taking part 1 if their knowledge or experience level permits).

The tutorial offerings are usually in high demand, and some sell out before pre-registration closes. Attendance is limited, and pre-registration is strongly recommended.

Track 1	Track 2	Track 3
Networking Part 1	Intro: PERL	Intro: Sys Admin Part 1 <i>Monday AM</i>
X Admin Part 1	Domain Name System	Intro Sys Admin Part 2 <i>Monday PM</i>
Networking Part 2	Advanced PERL	NEW Topics Part 1 <i>Tuesday AM</i>
X Admin Part 2	Sendmail + IDA sendmail	NEW Topics Part 2 <i>Tuesday PM</i>

Introductory System Administration - Part 1:

This half-day of intermediate material covers everything you need to know about logins (creating users and manipulating the administration files) and backups (including short descriptions of the various commercial heterogeneous backup solutions). Additionally, the session includes an introduction to the problems of security at your site and the COPS security analysis system.

Introductory System Administration - Part 2:

This half-day of intermediate material covers setup and operation of C news; setup and opera-

ation of your machine room; and setup and operation of the UUCP package for connecting your computer to the outside world.

Networking – Part 1:

This first half of the networking track includes an overview of networking and how it works; a description of how packets are switched throughout the internet; an introduction to transporting packets around your site via routers, bridges, and gateways; and a discussion of the new high speed modems and how they can foster fast, inexpensive communication.

Networking – Part 2:

The second half of the networking track concentrates administration of users on a network. It includes discussions of the Network Filesystem and its configuration in addition to the use of automounters to reduce administrative overhead on medium and large networks. The last part of the day discusses SLIP, a scheme for using serial lines as a low to medium speed network connectivity tool.

NEW Topics in System Administration – Part 1:

The popular "Topics in System Administration Series" continues with all new material for 1993. The first half discusses site maintenance using *rdist* for shuttling files among many systems, how to organize filesystems in large, heterogeneous environments, source tree management for multiple architectures, quick configuration and installation of workstations, and accounting.

NEW Topics in System Administration – Part 2:

The second of the the all new material includes: use of daemons to increase privileges of non-root users, trouble management systems, text processing previewers, console concentrators, NNTP (the network news transfer program which can reduce netnews traffic on your LAN), maintenance of large mail gateways, and electronic mail privacy.

X and the Administrator – Part 1:

This tutorial is targeted at system administrators who already know how to use X, but want to learn more about what goes on "behind the scenes." It includes an overview of the different components that make up X Windows (server, clients, different vendor products, etc.). We discuss where the files required to run X are usually located and what they do. We also discuss in detail how to configure a user's environment (e.g., all the different "dot" files and environment

variables). We then cover how to administer X terminals and what to look for when buying an X terminal. Finally, we discuss the tasks involved in maintaining the X source code distribution from MIT. There is also a troubleshooting section which includes hints and tips for resolving problems.

X and the Administrator – Part 2:

This tutorial builds on the concepts learned in part 1 (or through experience administering X) and includes everything you need to know about fonts: useful utilities, converting between different font formats, and using the X11R5 font server. We include discussions on using *imake* and how to manage multiple versions of X. We discuss some of the security issues associated with X and what you can do to deal with these issues. We also examine how to manage X in a distributed environment with multiple server and host types. Finally, we conclude with some advanced hints and tips for troubleshooting.

Configuring Sendmail:

This session will concentrate on modifying, programming, and debugging sendmail configuration files. Not only will syntax and semantics be covered but also test and verification techniques. The extended time will allow examination of several exemplary pieces of configuration files and a complete explication of testing and verifying sendmail configuration files – including a verification suite.

IDA Sendmail:

IDA Sendmail is a net-supported, rapidly evolving version of sendmail originally based on 4.3 BSD sendmail. It gives the administrator the flexibility of direct access to dbm files (among other things) and comes ready to install "as is" on almost any system. You may want to consider IDA Sendmail as the "total sendmail solution" for your site. This talk covers the IDA sendmail specifics -- not the general problem of configuring sendmail for your site.

The Domain Name System:

DNS, the Domain Name System, is a distributed database to handle hostname to IP address look-ups and to help in routing mail. This session includes a look at how it arose, the problems of scale it was trying to solve, how to configure it, routine maintenance and debugging. We detail how to set up includes, establishing primary server configuration, using tools for maintaining the forward and reverse files, configuring a resolver, handling MX records, and a bit about

designing a robust name service scheme for your organization.

Introduction to Perl Programming:

Perl is a publicly available and highly portable interpreted programming language occupying the large niche between shell and C programming, and as such is excellent for many system management tasks. This tutorial is suitable for individuals who have never looked at Perl before or have only just begun to use it. Students with a background in UNIX shell programming and regular expressions will benefit most from this course. Topics of this tutorial include detailed descriptions and numerous examples of the syntax and semantics of the language, its data types, operators, control flow, regular expressions, and I/O facilities, and using the Perl debugger.

Advanced Perl Programming:

This brand-new course is designed for programmers already experienced with Perl who would like to expand their Perl expertise about sophisticated data types, complex networking, and advanced code conversion. Students with a firm background in both Perl and UNIX C programming will benefit most from this course. Topics of this tutorial include packages to create your own libraries, using pointers to synthesize complex data types (such as list of lists or arrays of records), the bit vector data type and the *select()* system call, using *h2ph* and *c2ph* to convert and access C code, socket programming, the *ioctl* and *fcntl* system calls, and exception handling.

The Instructors:

Tom Christiansen, *Convex Computer Corp.*
Trent Hein, *XOR Computer Systems*
Dr. Rob Kolstad, *Berkeley Software Design, Inc.*
Dinah McNutt, *Tivoli Systems*
Dr. Evi Nemeth,
University of Colorado at Boulder
Miles O'Neal, *Pencom Software*
Jeff Polk, *Berkeley Software Design, Inc.*

Technical Program

**Wednesday - Friday
October 21-23**

At press time, the LISA VI abstract deadline just passed. The program committee is currently reviewing over 50 proposals for papers on a variety of topics. We have received a good range of papers, covering most of the topics that were suggested in the Call For Papers, as well as additional ones. The committee is very pleased with the response to the Call and is looking forward to presenting a strong technical program at LISA VI. Here are some of the topics that are likely to be included in the program:

- Tools for Real-Time System Troubleshooting
- Tricks in User Education
- Graphical User Interfaces for System Administration
- Distributed System Administration
- Experiences Using Third-party Administration Software
- Network Growth and Performance Management
- System Security Monitoring
- Evaluating Performance of High-End Workstations and Servers
- Keys to Successful, Painless Upgrades
- Object Management Systems for System Administration
- Standardization of System Administration
- Heterogeneous System Administration
- System Archiving and Backups

If you would like to host a BOF session, have a suggestion for an alternate track, or need information about how your company can participate in the vendor display, please contact the program chair (see below).

Complete information about registration and hotels will be mailed to the membership in August.

Contact Information:

Trent Hein, Program Chair
XOR Computer Systems
2525 Arapahoe, Suite E4-264
Boulder, Colorado 80302
(303) 440-6093
<trent@xor.com>

UNIX Applications Development Symposium

Call for Participation: Toronto, Ontario, Canada March 29 - April 1, 1993

Co-sponsored by the USENIX Association and UniForum Canada.

One of the major uses of UNIX today is the support, development, and execution of applications ultimately used in achieving end users' business goals. The current trends in large end-user organizations of downsizing major applications from older mainframes to less expensive, more powerful, and simpler, modern networked, machines lend UNIX a serious position in the commercial marketplace. Consequently, more and more computing and information systems professionals are encountering UNIX when developing and maintaining applications.

The purpose of this symposium is to expose the challenges of building and maintaining applications on UNIX platforms, to discuss solutions and experiences, and to explore existing practice and techniques.

This symposium will feature papers, invited talks, panel discussions, and tutorials on aspects of designing, building, testing, debugging, and maintaining applications within and for the UNIX environment. There will also be ample opportunity at this symposium to meet your peers and make contact with others with similar interests.

This symposium will provide valuable information to designers, programmers, and managers who are planning to port existing applications into the UNIX environment or move development and maintenance teams from proprietary environments to UNIX.

Important Dates for Refereed Paper Submissions

Extended Abstracts Due:	December 4, 1992
Notifications to Authors:	December 16, 1992
Final Papers Due:	February 12, 1993

Other Important Dates

Pre-registration materials will be available in mid-January, 1993

Tutorial Program

Tues., March 29, 1993

Technical Sessions

Wed., March 30 - Thu., April 1, 1993

Birds-Of-a-Feather Sessions

Tues., March 29 - Thu., April 1, 1993

USENIX Reception

Thurs. evening, April 1, 1993

USENIX is co-sponsoring this event with UniForum Canada, a non-profit membership organization.

Suggested Topics:

Topics may include, but are not limited to:

Graphical User Interfaces - The X Window System - User Interface Design & Standards. Open Look, Motif, NeWS, and so on. What is a style guide? Importance of consistency and ease of use.

Porting Issues - Issues surrounding the tasks of porting an existing application to UNIX, as well as issues of making UNIX applications portable to other architectures and other platforms.

Networking - Client/Server design issues, etc.

Project Management - Using UNIX tools to support project management. CASE - What, When, Why, Who, How.

O/S Issues - Overcoming limitations set by hardware and operating systems.

Security - The impact of security features. Schemes for maintaining security within an application.

Transaction Processing - Implementing distributed transaction processing for UNIX applications.

Fourth Generation Languages - What advantages and disadvantages do 4GL's have in a UNIX environment?

Distributed Applications - How do you make the best use of existing UNIX functionality (such as e-mail) to build UNIX applica-

tions? What are the issues of building and/or using distributed databases?

Object Oriented Programming - Productivity, languages, techniques, case studies, etc.

Object Oriented Databases - Advantages, etc.

The Corporate Internet - High Speed for the Elite, or Connectivity for the Masses? ISDN, TCP/IP, OSI, UUCP. Governments, privateers, service providers, co-operatives, telecoms. Network philosophy - open road, tollbooths, freeloaders or lifeblood.

Delivering/Installing Applications - What's the best way? How to prevent piracy, worms, viruses, etc. How to do updates effectively and securely.

Testing & Certifying Binary Applications - Who does this? What does this achieve? How long does it take? Applications and POSIX.1 Conformance Testing.

Standards - ABI/API/ANDF - How, What, Where, When, Why? What are they? How are these standards used? How do they affect applications? What features does each have? What benefits are derived from using each? Where should they be used/followed? When will they be real? How do you keep up with new standards? Why are they necessary?

Submission Details

Papers may feature real-life experiences, as well as research topics. Both case-study and technical papers will be accepted. Case studies should describe existing systems and include implementation details and may also include performance data where practical.

Submissions must be in the form of extended abstracts (1500-2500 words; 3-5 pages in length). Shorter abstracts might not give the program committee enough information to judge your work fairly and, in most cases, your submission will be rejected. Longer abstracts and full papers simply cannot be read by the committee in the time available. Feel free to append a full paper to an extended abstract; this is sometimes useful during evaluation. The extended abstract should represent your paper in *short form*. The committee wants to see that you have a real project, that you are familiar with the work in your area, and that you can clearly explain yourself.

Please note that presentations are usually scheduled to last 25 minutes. Your presentation should provide an overview of your paper and entice your audience to read it in the proceedings and

hopefully follow up on your solution, or take your advice into consideration.

Papers will be judged on technical merit, relevance to the theme, and suitability for presentation. Papers are welcome from software (and hardware) vendors who wish to share their innovative solutions and techniques, but be forewarned that product marketing will not be tolerated.

Persons interested in participating in panel discussions should contact <woods@usenix.org>.

Tutorial Program

Tutorial Coordinator: Dan Klein
<dvk@usenix.org> Tel: 412-421-2332

Explore topics essential to successful use and development of UNIX and UNIX-like operating systems, X windows, networking and interoperability, advanced programming languages, and related areas of interest. The USENIX Association's well-respected tutorial program offers you introductory and advanced, intensive yet practical tutorials. Courses are presented by skilled teachers who are hands-on experts in their topic areas.

In an effort to continue to provide the best possible tutorial slate, USENIX is soliciting proposals for new tutorials. If you are interested in presenting a tutorial, contact the Tutorial Coordinator (see above).

Invited Talks

Interim Invited Talks and Panel Co-ordinator:
Greg Woods <woods@usenix.org>

As part of the technical sessions, a series of invited talks provides introductory and advanced information about a variety of interesting topics, such as using standard UNIX tools and employing specialized applications. We welcome suggestions for topics as well as request proposals for particular talks. In your proposal, state the main focus, include a brief outline, and be sure to emphasize why your topic is of general interest to our community.

Birds-of-a-Feather Sessions

BOF Scheduling: USENIX Conference Office
<conference@usenix.org>

Birds-of-a-Feather sessions (BoFs) bring together devotees of many varied disciplines for discussions, announcements, mingling, and strategy sharing during evenings at the symposium. Schedule a BoF in advance or on-site.

Work-in-Progress Reports

WIPS Coordinator: Greg Woods
<woods@usenix.org>

These reports provide researchers with 10 minutes to speak on current work and receive valuable feedback. Present your interim results, novel approaches, or newly-completed work. Schedule your report in advance or on-site.

For More Information

Materials containing all details of the technical and tutorial program, conference registration, hotel and airline discount and reservation information will be mailed in January of 1993.

For further information about the Symposium, contact the Program Chair:

Program Chair

Greg A. Woods
#3 - 46 Three Valleys Drive
Don Mills, Ontario
M3A 3B5, CANADA
<woods@usenix.org>
+1 416 443-1734
+1 416 595-5425 [FAX]

Current Program Committee:

Rob Kolstad, Berkeley Software Design
<kolstad@bsd.com>
Evan Leibovitch, Sound Software
<evan@telly.on.ca>
Peter Renzland, Ontario Government
<peter@renzland.org>
Dan Tomlinson, Compusoft
<compus!dan@uunet.uu.net>
Greg Woods, Elegant Communications
<woods@usenix.org>
Elizabeth Zwicky, SRI International
<zwicky@erg.sri.com>

Mach Symposium

Preliminary Announcement & Call for Papers Santa Fe, NM April 19-21, 1993

Extended Abstracts Due: December 4, 1992

Background

The use and influence of Mach on the operating systems community continues to grow. From its beginnings as a small research project, Mach has spread to become the basis for commercial products from a variety of vendors, and a key component of innovative research efforts in both academic and industrial environments. At the same time, research and development continue to evolve Mach itself. The community of researchers and developers working with Mach is proving to be a very productive source of innovative systems.

Activity in this field has been sufficiently widespread that the USENIX Association is pleased to once again sponsor a Mach symposium to bring

together researchers, engineers, vendors and users of Mach systems. We will encourage discussion of all past and present Mach-related research, development, production, and applications activities.

Symposium Overview

The symposium will be spread over three days. The first day will be devoted to tutorials on Mach 3.0, and will include both introductory/overview and advanced programming tracks. These tutorials should be of interest to both those desiring an introduction to Mach, and to programmers interested in learning how to take better advantage of Mach features. The following two days will concentrate on presentation of refereed papers on past and present Mach-related work. Long breaks between presentations will provide opportunities for informal discussion. Some time will be available for descriptions of work in progress.

Extended abstracts of 1500-2500 words (9000-15000 bytes or 3-5 pages) should be sent to David Black at the address below (those submitting hardcopy abstracts must send five copies). Shorter abstracts run a significant risk of rejection as there will be little on which the program committee can base an opinion. In addition to the extended abstract, authors must also supply an outline of the full paper and an estimate of its length.

A good extended abstract will contain the following information in one form or another:

Abstract	100-300 words (half a page) included verbatim in the final paper
Introduction	The problem; its importance; previous work
Solution	Issues, decisions, tradeoffs, rationale
Implementation details	
Evaluation	Performance results; effort required; lessons learned.

Conclusion

The extended abstract allows the program committee to analyze the content of the proposed paper. An outline lists the headings, major points, and many minor points for each section of the actual paper.

The outline should provide an idea of the form and style of your paper. This layout is not cast in concrete; just submit enough material to convince the committee that they want to accept the paper!

Longer abstracts (up to and including full papers) will be reviewed; the program committee appreciates any additional material that makes it easier to predict the content, organization, and style of the final paper. Authors should exercise appropriate restraint in determining the amount of material to submit.

The submission package must include:

- The extended abstract
- Outline of rest of paper
- Cover letter, detailing:
 - Title of paper
 - Authors
 - Estimate of paper length
 - Contact author (liaison to program committee)
 - E-mail address and daytime phone number for contact author
 - Hours during which the daytime phone number can be used
 - Surface mail address
 - Optional FAX and home phone numbers

If hardcopy is being submitted, five copies of the submission.

The submission should be sent electronically to *dlb@osf.org*, or by surface mail (five copies of abstract) to David Black at the address listed

below. Submissions made by FAX will not be accepted.

Electronic submissions must be in plain text that can be reviewed in the form submitted; the program committee does not have the time to run formatting tools (e.g., TeX, LaTeX) or to figure out why a printer refuses to print some PostScript document.

All submissions will be acknowledged. Authors of approved abstracts will be required to submit full-length papers (8-15 pages) approximately five weeks after notification of acceptance. Formatting guidelines will be provided.

Areas of interest include, but certainly are not limited to:

- Applications and support for programming languages
- Mach 2.5 and related systems (e.g., OSF/1)
- Mach 3.0 and servers
- Mach-based operating system implementation and emulation
- Use of Mach subsystems in other operating systems
- Multiprocessor and parallelization experience
- Distributed systems, including multicomputers, clusters, etc.
- Real Time
- Security
- Performance
- Productization experiences
- Comparisons of Mach with other operating systems (e.g., Chorus, Sprite, Amoeba, V, and of course, UNIX)
- Future work

The program committee is especially interested in papers describing applications and/or system servers that take advantage of Mach features in addition to papers describing the evolution of Mach kernel technology. Submissions are strongly encouraged from efforts across the entire spectrum from research projects to product development efforts (including work that falls between these endpoints).

Important dates:

Extended abstracts:	December 4, 1992
Notification to Authors:	January 18, 1993
Camera-ready, full papers:	February 26, 1993

For further information about the symposium, contact the program chair at the address on the following page.

Program Chair

David Black
Research Institute
Open Software Foundation
1 Cambridge Center, 11th Floor
Cambridge, MA 02142
Voice: +1 (617) 621-7347
FAX: +1(617) 621-8696
E-Mail: dlb@osf.org

Program Committee

David Black, *Open Software Foundation*
David Golub, *Carnegie Mellon University*
Alan Langerman, *Orca Systems, Inc.*
Jay Lepreau, *University of Utah*
Avadis Tevanian, Jr., *NeXT, Inc.*

EurOpen Spring '93 Conference

Call for Papers Seville, Spain May 3 - 7, 1993

The UUES (Spanish UNIX User Group) will host the 24th EurOpen Conference and Exhibition in Seville, Spain, on May 3-7, 1993.

The event is centered on a conference, exhibition and associated tutorials, dedicated to UNIX and Open systems.

The three day conference with local commercial Exhibition will take place on May 5-7. It will be preceded by two days of Tutorials.

A pre-conference registration pack containing detailed information will be available early in 1993.

The theme of the Seville Conference is: "Open Systems from the desktop to the machine room: the new challenge."

Many believe that UNIX can provide a single open environment for machines from desktop size to mainframes and super-computers. However, it is uncertain that UNIX will sustain its current moderate penetration in the desktop market, and some commercial DP and MIS departments have been slow to adopt UNIX. Remaining sufficiently open presents a new challenge to the UNIX community.

One of the attractions of UNIX is reputedly the ability to easily migrate applications between various hardware bases. This includes the possibility of migrating from large centralised systems across to distributed client-server environments, includ-

ing desktop support. Once again, openness and portability are critical issues.

Within the UNIX community itself, there is significant interest in the possibility of small micro-kernels providing a flexible replacement for monolithic UNIX environments. Such flexibility may be the key to providing growth from small systems to very large ones, particularly multi-computers. They may also be a basis towards openness for non-UNIX operating systems by providing a multi-faceted environment within a single machine.

A consequence of openness in approach is that users can acquire and understand UNIX at limited cost. In practice this not only results from using a desktop, rather than a mainframe machine, but also from the availability of public domain as well as proprietary software, and low cost UNIX implementations. Training and support must likewise be open, in the sense that they must be sufficiently flexible to meet the highly varied requirement of end users.

Important Dates

Deadline for receipt of full papers, or extended abstracts, by the Secretariat: October 25, 1992

Notification to authors of the Programme Committee's decision: November 29, 1992

Deadline for receipt of the final paper: January 29, 1993

Submission of Papers

Paper submissions should identify the author(s) and the organisation(s) to which they belong. A

submission should include a draft of the complete paper (5 to 10 pages), or at least an extended abstract (at least 2 pages). Each will be examined on the basis of its originality, clarity, technical quality, and adherence to the general theme of the conference.

The Programme Committee wishes to include both technical papers and syntheses of different approaches. The quality of the Conference depends to a large extent on the quality of the presentations and papers: authors are thus strongly encouraged to supply a complete version of their text as soon as possible.

Submissions should be sent direct to the EurOpen Secretariat (address below).

Tutorials:

The tutorials will provide attendees with information on specific topics. Their purpose will be to present the state of the art in important areas of open systems. The tutorials will be led by experts of national and international fame. Those interested in offering a tutorial should contact the EurOpen Tutorial Executive as soon as possible.

Useful Addresses

Secretariat

EurOpen Secretariat
Owles Hall
Buntingford
UK - Herts SG9 9PL
Telephone: +44 763 73039
Facsimile: +44 763 73255
Email: europen@EU.net

Program Chair

Dr. Chris Horn
Iona Technologies
Innovation Center
Trinity College
Dublin 2 Ireland
Tel: + 353 1 679 0677
Facsimile: + 353 1 679 8039
Email: horn@iona.ie

Tutorial Executive

Neil Todd
GID Ltd.
1 Captains Close
Upper Basildon
Reading, Berks RG8 8SZ
UK
Telephone: + 44 491 671 964
Facsimile: + 44 763 73255
Email: neil@pio.gid.co.uk

Publications Order Form

Subscription to Proceedings

Please enter my one-year subscription* to the 1992 USENIX Conference, Workshop and Symposia Proceedings, which include:

San Francisco Technical Conference - January 1992

Symposium on Experiences with Distributed & Multiprocessor Systems (SEDMS) III - March 1992

Workshop on Micro-Kernels and other Kernel Architectures - April 1992

File Systems Workshop - May 1992

San Antonio Technical Conference - June 1992

C++ Conference - August 1992

UNIX Security III Symposium - September 1992

Systems Administration (LISA) VI Conference - October 1992

***NOTE:**

Corporate, Educational and Supporting Members of USENIX automatically receive a subscription to all proceedings published during their term of membership, which may overlap with this offer.

COST: USENIX members: \$170.00

Non-members: \$214.00

Outside U.S.A. and Canada?

Add \$30.00 for surface postage.

Subscription Cost _____

**

Calif. Sales Tax _____

Overseas Postage _____

TOTAL ENCLOSED _____

** If you are paying member price, please include member's name and/or membership number _____

PAYMENT OPTIONS

☐ Check enclosed payable to USENIX Association. ☐ Purchase order enclosed.

☐ Please charge my: ☐ Visa ☐ MasterCard



Account # _____ Exp. Date _____

Signature _____

Outside the U.S.A.? Please make your payment in U.S. currency by one of the following:

- * Check - issued by a local branch of a U.S. Bank
- * Charge (Visa, MasterCard, or foreign equivalent)
- * International postal money order

4/92

Name _____

Address _____

City _____ State/Country _____ Zip/Postal Code _____

Please mail this order form to: USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
510/528-8649
FAX 510/548-5738
office@usenix.org

This offer is good through
December 31, 1992

The proceedings are shipped the week following the event.

Publications Order Form

Conference & Workshop Proceedings

Qty	Proceedings	Member Price	Non-Member Price	Subtotal	Overseas Postage	Total
WINTER/SUMMER CONFERENCES						
___	San Antonio Summer '92	\$23	\$30	\$ _____	\$14	\$ _____
___	San Francisco Winter '92	30	39	\$ _____	22	\$ _____
___	Nashville Summer '91	32	38	\$ _____	22	\$ _____
___	Dallas Winter '91	28	34	\$ _____	18	\$ _____
___	Anaheim Summer '90	22	22	\$ _____	15	\$ _____
___	Washington, DC Winter '90	25	25	\$ _____	15	\$ _____
___	Baltimore Summer '89	20	20	\$ _____	15	\$ _____
___	San Diego Winter '89	30	30	\$ _____	20	\$ _____
___	San Francisco Summer '88	29	29	\$ _____	20	\$ _____
___	Dallas Winter '88	26	26	\$ _____	15	\$ _____
___	Phoenix Summer '87	35	35	\$ _____	20	\$ _____
___	Washington, DC Winter '87	10	10	\$ _____	15	\$ _____
___	Atlanta Summer '86	37	37	\$ _____	20	\$ _____
___	Denver Winter '86	25	25	\$ _____	15	\$ _____
___	Portland Summer '85	45	45	\$ _____	25	\$ _____
___	Dallas Winter '85	15	15	\$ _____	10	\$ _____
___	Salt Lake City Summer '84	29	29	\$ _____	20	\$ _____
___	Washington, DC Winter '84	25	25	\$ _____	15	\$ _____
___	Toronto Summer '83	32	32	\$ _____	20	\$ _____
___	San Diego Winter '83	28	28	\$ _____	15	\$ _____
LARGE INSTALLATION SYSTEMS ADMINISTRATION						
___	LISA V Sept. '91	20	23	\$ _____	11	\$ _____
___	LISA IV Oct. '90	15	18	\$ _____	8	\$ _____
___	LISA III Sept. '89	13	13	\$ _____	9	\$ _____
___	LISA II Nov. '88	8	8	\$ _____	5	\$ _____
___	LISA I April '87	4	4	\$ _____	5	\$ _____
C++						
___	C++ Conference Aug. '92	30	39	\$ _____	20	\$ _____
___	C++ Conference Apr. '91	22	26	\$ _____	11	\$ _____
___	C++ Conference Apr. '90	28	28	\$ _____	18	\$ _____
___	C++ Conference Oct. '88	30	30	\$ _____	20	\$ _____
___	C++ Workshop Nov. '87	30	30	\$ _____	20	\$ _____
SECURITY						
___	UNIX Security II Aug. '90	13	16	\$ _____	8	\$ _____
___	UNIX Security Aug. '88	7	7	\$ _____	5	\$ _____
MACH						
___	Mach Symposium Nov. '91	24	28	\$ _____	14	\$ _____
___	Mach Workshop Oct. '90	17	20	\$ _____	9	\$ _____

Continued - see reverse

Qty	Proceedings		Member Price	Non-Member* Price	Subtotal	Overseas Postage	Total
DISTRIBUTED & MULTIPROCESSOR SYSTEMS (SEDMS)							
___	SEDMS III	Mar. '92	30	36	\$ _____	20	\$ _____
___	SEDMS II	Mar. '91	30	36	\$ _____	20	\$ _____
___	SEDMS	Oct. '89	30	30	\$ _____	20	\$ _____
GRAPHICS							
___	Graphics Workshop V	Nov. '89	18	18	\$ _____	10	\$ _____
___	Graphics IV	Oct. '87	10	10	\$ _____	10	\$ _____
___	Graphics III	Nov. '86	10	10	\$ _____	5	\$ _____
___	Graphics II	Dec. '85	7	7	\$ _____	5	\$ _____
OTHER WORKSHOPS							
___	File Systems	May '92	15	20	\$ _____	9	\$ _____
___	Micro-Kernel & Other Kernel Arch.	April '92	30	39	\$ _____	20	\$ _____
___	UNIX Transaction Processing	May '89	12	12	\$ _____	8	\$ _____
___	Software Management	Apr. '89	20	20	\$ _____	15	\$ _____
___	UNIX & Supercomputers	Sept. '88	20	20	\$ _____	10	\$ _____



Discounts are available for bulk orders. Please inquire.

Total price of Proceedings _____ **
Calif. residents add sales tax _____
Total overseas postage _____
Total enclosed _____

**If you are paying member price, please include member's name and/or
 membership number _____

PAYMENT OPTIONS*

___ Check enclosed - payable to USENIX Association

___ Charge my: ___ VISA  ___ MC  Account # _____ Exp. Date _____

___ Purchase order enclosed Signature _____

* Outside the USA? Please make your payment in US currency by one of the following:

- Check - issued by a local branch of a US Bank
- Charge (VISA, MasterCard, or foreign equivalent)
- International postal money order

Shipping Information

Ship to: _____

Please allow 2 - 3 weeks for delivery. Overseas orders are shipped via air printed matter.

Please mail or fax this order form with payment to:

USENIX Association
 2560 Ninth Street, Suite 215
 Berkeley, CA 94710
 FAX 510/548-5738

• If you are not a member and wish to receive our membership information packet, please check this box. ☐

Local User Groups

The Association will support local user groups by doing a mailing to assist in the formation of a new group and publishing information on local groups in ;login:. At least one member of the group must be a current member of the Association. Send additions and corrections to:

login@usenix.org.

CA - Fresno:

The Central California UNIX Users Group consists of a uucp-based electronic mailing list to which members may post questions or information. For connection information:

Educational and governmental institutions:

Brent Auernheimer (209) 278-2573

brent@CSUFresno.edu or *csufres!brent*

Commerical institutions or individuals:

Gordon Crumal (209) 251-2648

csufres!gordon

CA - Orange County:

Meets the 2nd Monday of each month

UNIX Users Association of Southern California

Paul Muldoon (714) 556-1220 ext. 137

New Horizons Computer Learning Center

1231 E. Dyer Rd., Suite 140

Santa Ana, CA 92705

CO - Boulder:

Meets monthly at different sites. For meeting schedule, send email to *fruug-info@fruug.org*.

Front Range UNIX Users Group

Software Design & Analysis, Inc.

1113 Spruce St., Ste. 500

Boulder, CO 80302

Steve Gaede (303) 444-9100

gaede@fruug.org

FL - Coral Springs

S. Shaw McQuinn (305) 344-8686

8557 W. Sample Road

Coral Springs, FL 33065

FL - Western:

Meets the 1st Thursday of each month.

Florida West Coast UNIX Users Group

Richard Martino (813) 536-1776

Tony Becker (813) 799-1836

mcrsys!tony

Ed Gallizzi, Ph.D. (813) 864-8272

e.galizzi@compmail.com

Jay Ts (813) 979-9169

uunet!pdn!tscs!metran!jan

Dave Lewis (407) 242-4372

dhl@ccd.harris.com

FL - Orlando:

Meets the 3rd Thursday of each month.

Central Florida UNIX Users Group

Mike Geldner (407) 862-0949

codas!sunfla!mike

Ben Goldfarb (407) 275-2790

goldfarb@hcx9.ucf.edu

Mikel Manitijs (407) 869-2462

[codas,attmail]!mikel

FL - Melbourne

Meets the 3rd Monday of every month.

Space Coast UNIX User's Group

Steve Lindsey (407) 242-4766

lindsey@vnet.ibm.com

KS or MO - Kansas:

Meets on 2nd Monday of each month.

Kansas City UNIX Users Group (KUUG)

813B St.

Blue Springs, MO 64015

(816) 235 5212

mlg@cstp.umkc.edu

GA - Atlanta:

Meets on the 1st Monday of each month in White Hall, Emory University.

Atlanta UNIX Users Group

P.O. Box 12241

Atlanta, GA 30355-2241

Mark Landry (404) 365-8108

MI - Detroit/Ann Arbor:

Meets on the 2nd Thursday of each month in Ann Arbor.

Southeastern Michigan Sun Local Users Group

and Nameless UNIX Users Group

Steve Simmons office: (313) 769-4086

home: (313) 426-8981

scs@lokkur.dexter.mi.us

MI - Detroit/Ann Arbor (cont.)

K. Richard McGill
rich@sendai.ann-arbor.mi.us
Bill Bulley
web@applga.uucp

MN- Minneapolis/St. Paul:

Meets the 1st Wednesday of each month.

UNIX Users of Minnesota
17130 Jordan Court
Lakeville, MN 55044
Robert A. Monio (612) 220-2427
pnessutt@dmshq.mn.org

MO - St. Louis:

St. Louis UNIX Users Group
P.O. Box 2182
St. Louis, MO 63158
Terry Linhardt (314) 772-4762
uunet!jgaltstl!terry

NE - Omaha:

Meets monthly.

/usr/group/nebraska
P.O. Box 31012
Omaha, NE 68132
Phillip Allendorfer (402) 423-1400

New England - Northern:

Meets monthly at different sites.

Peter Schmitt 603) 646-2085
Kiewit Computation Center
Dartmouth College
Hanover, HN 03755
Peter.Schmitt@dartvax!dartmouth.edu

NJ - Princeton:

Meets monthly.

Princeton UNIX Users Group
Mercer County Community College
1200 Old Trenton Road
Trenton, NJ 08690
Peter J. Holsberg (609) 586-4800
mccc!pjh

NY - New York City:

Meets every other month in Manhattan.

Unigroup of New York City
G.P.O. Box 1931
New York, NY 10116
Peter Gutmann (212) 618-0973 *peterg@murphy.com*

OK - Tulsa:

Meets 2nd Wednesday of each month.

Tulsa UNIX Users Group, \$USR
Stan Mason (918) 560-5329
tulsix!smason@drd.com
Mark Lawrence (918) 743-3013
mark@drd.com

TX - Austin:

Meets 3rd Thursday of each month.

Capital Area Central Texas UNIX Society
P.O. Box 9786
Austin, TX 78766-9786
officers@cactus.org
Tom Painter (512) 835-5457
president@cactus.org

TX - Dallas/Fort Worth:

Dallas/Fort Worth UNIX Users Group
660 Preston Forest, Suite 177
Dallas, TX 75230
Kevin Coyle (214) 991-5512
kevincd@shared.com

TX - Houston:

Meets 3rd Tuesday of each month.

Houston UNIX Users Group
(Hounix) answering machine (713) 684-6590
Bob Marcum, President (713) 270-8124
Chuck Bentley, Vice-president
(713) 789-8928 *chuckb@hounix.uucp*

WA - Seattle:

Meets monthly.

Seattle UNIX Group Membership Info.
Bill Campbell (206) 947-5591
6641 East Mercer
Mercer Island, WA 98040-0820
bill@celestial.com

Washington, D.C.:

Meets 1st Tuesday of each month.

Washington Area UNIX Users Group
9811 Mallard Drive
Laurel, MD 20708
Alan Fedder (301) 953-3626

CANADA - Toronto:

143 Baronwood Court
Brampton, Ont. Canada L6V 3H8
Evan Leibovitch (416) 452-0504
evan@telly.on.ca

Calendar of Events

1992

- Sep 8-11 AUUG, Melbourne, Australia
14-17 * UNIX Security III, Baltimore, MD
22-24 UNIX Expo, New York, NY
Oct 6 ISO/IEC JTC1 SC22 WG15, Denmark
18-22 OOPSLA, Vancouver, Canada
19-23 * LISA VI, Long Beach, CA
19-23 IEEE 1003, Utrecht, Netherlands
26-30 Interop, San Francisco, CA
27 - 30 ISO/IEC JTC1 SC22 WG15
Reading, UK
Nov 25-27 EurOpen/UniForum
Utrecht, Netherlands
Dec 7 Sun User Group, San Jose, CA
UKUUG/UKnet, Manchester, UK

1993

- Jan 11-15 IEEE 1003, New Orleans, LA
25-29 * USENIX, San Diego, CA
Feb 22-24 Sun Open Sys. Expo, Chicago, IL
Mar 15-18 UniForum, San Francisco, CA
31 - * UNIX Applications Development
Apr 1 Toronto, Canada
19-21 * Mach, Santa Fe, NM
19-23 IEEE 1003
May 3 - 7 EurOpen, Seville, Spain
Jun 21-25 * USENIX, Cincinnati, OH
Jul 12-16 IEEE 1003
Autumn EurOpen/UniForum
Utrecht, Netherlands
* LISA VII
* SEDMS
* Micro-Kernel
Oct 18-22 IEEE 1003
25-29 Interop, San Francisco, CA

1994

- Jan 17-21 * USENIX, San Francisco, CA
Feb 14-17 UniForum, Dallas, TX
Mar 23-25 UniForum, San Francisco, CA
Apr 18-22 EurOpen

- Jun 6-10 * USENIX, Boston, MA
Sep 12-16 Interop, San Francisco, CA
Autumn EurOpen/UniForum
Utrecht, Netherlands

1995

- Jan 16-20 * USENIX, New Orleans, LA
Feb 21-23 UniForum, Dallas, TX
May 1- 5 EurOpen
Jun 19-22 * USENIX, San Francisco, CA

1996

- Jan 22-26 * USENIX, San Diego, CA
Mar 11-14 UniForum, San Francisco, CA

This is a combined calendar of planned conferences, workshops, and standards meetings related to the UNIX operating system. If you have a UNIX-related event that you wish to publicize, please contact login@usenix.org. Please provide your information in the same format as above. This calendar has been compiled with the assistance of Alain Williams of EurOpen.

* = events sponsored by the USENIX Association.

ACE: Advanced Computing Environments
ACM: Association for Computing Machinery
AFUU: Association Francaise des Utilisateurs d'UNIX

AUUG: Australian UNIX Users Group
DECUS: Digital Equipment Computer Users Society
ECUG: European C++ User Group

EurOpen: European Forum for Open Systems
GUUG: German UNIX Systems User Group
IEEE: Institute of Electrical and Electronics Engineers

IETF: Internet Engineering Task Force
Interex: Internat'l Assoc. - Hewlett-Packard Comp. Users
JUS: Japan UNIX Society

LISA: USENIX Systems Administration Conference
NIST: National Institute of Standards & Technology
SEDMS: Symposium on Experiences with Distributed and Multiprocessor Systems

Sinix: Singapore UNIX Association
UKUUG: United Kingdom UNIX Systems Users Group

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

**Second Class Postage
PAID
At Berkeley, California
and Additional Offices**

What's Inside?

Special Technical Groups

New Section: SAGE

Micro-Kernel Workshop Report

Standards Activity Update

The Bookworm